NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MOBILE SENSOR NETWORKS: A DISCRETE EVENT SIMULATION OF WMD THREAT DETECTION IN URBAN TRAFFIC SCHEMES

by

Jeffrey F. Hyink

March 2007

Thesis Advisor:	A. H. Buss
Second Reader:	P. J. Sanchez

Approved for public release; distribution unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2007	**3. REPORT TYPE AND DATES COVERED** Master’s Thesis

4. TITLE AND SUBTITLE Mobile Sensor Networks: A Discrete Event Simulation of WMD Threat Detection in Urban Traffic Schemes

5. FUNDING NUMBERS

6. AUTHOR(S) CDR J. F. Hyink

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

The rise of the threat of WMD attack on American soil necessitates new and innovative approaches to homeland security. A layered security model has been proposed in which an attacker must successfully penetrate multiple defensive constructs in order to complete an attack. As part of a layered defensive approach, a network of sensor equipped vehicles operating in urban traffic is considered. To-date, sensor packages have been developed for vehicles without detailed, area-specific analysis of their aggregate performance measures. The possible effectiveness of this network of sensors in detecting vehicle based WMD attacks is explored in this thesis.

A Discrete Event Simulation using actual roadmap data was developed and analyzed to explore various configurations for searcher employment and in particular to generate a potential return on investment curve in the form of probability of detection generated as a function of the number of sensor equipped vehicles. The baseline scenario centers on an attacker utilizing a vehicle-mounted WMD device. The attacker attempts a shortest-path route from a randomly selected starting point to a downtown target node. Patrol vehicles are equipped with sensors that can identify potential attacker vehicles in the adjacent lane of oncoming traffic. These vehicles patrol the roadway network, and are assumed to foil an attack when they detect an attack vehicle. The simulation model outputs data such as the proportion of foiled attacks and the distance from target, given a detection.

An analysis of performance encompassing the greater Washington D. C. area to include over 620 square miles of urban and suburban roadway systems is conducted. Detector deployment in random search patterns in this roadway network yields an appreciable deterrent of greater than 10% probability of detection only when more than 200 patrolling agents are assigned. More optimized employment schemes, countermeasures, and counter-countermeasures are discussed in addition to other detection statistics and summary results.

14. SUBJECT TERMS mobile detectors, searcher detector model, GIS, GeoTools, Simkit, Washington D. C., Homeland Security, WMD detection, discrete event simulation

15. NUMBER OF PAGES
73

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

MOBILE SENSOR NETWORKS: A DISCRETE EVENT SIMULATION OF WMD THREAT DETECTION IN URBAN TRAFFIC SCHEMES

Jeffrey F. Hyink
Commander, United States Navy
B.S., Cornell University, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2007

Author: Jeffrey F. Hyink

Approved by: Professor A. H. Buss
Thesis Advisor

Professor P. J. Sanchez
Second Reader

James Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The rise of the threat of WMD attack on American soil necessitates new and innovative approaches to homeland security. A layered security model has been proposed in which an attacker must successfully penetrate multiple defensive constructs in order to complete an attack. As part of a layered defensive approach, a network of sensor equipped vehicles operating in urban traffic is considered. To-date, sensor packages have been developed for vehicles without detailed, area-specific analysis of their aggregate performance measures. The possible effectiveness of this network of sensors in detecting vehicle based WMD attacks is explored in this thesis.

A Discrete Event Simulation using actual roadmap data was developed and analyzed to explore various configurations for searcher employment and in particular to generate a potential return on investment curve in the form of probability of detection generated as a function of the number of sensor equipped vehicles. The baseline scenario centers on an attacker utilizing a vehicle-mounted WMD device. The attacker attempts a shortest-path route from a randomly selected starting point to a downtown target node. Patrol vehicles are equipped with sensors that can identify potential attacker vehicles in the adjacent lane of oncoming traffic. These vehicles patrol the roadway network, and are assumed to foil an attack when they detect an attack vehicle. The simulation model outputs data such as the proportion of foiled attacks and the distance from target, given a detection.

An analysis of performance encompassing the greater Washington D. C. area to include over 620 square miles of urban and suburban roadway systems is conducted. Detector deployment in random search patterns in this roadway network yields an appreciable deterrent of greater than 10% probability of detection only when more than 200 patrolling agents are assigned. More optimized employment schemes, countermeasures, and counter-countermeasures are discussed in addition to other detection statistics and summary results.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The Author would like to acknowledge the generous advice and assistance of Professors Arnold Buss and Paul Sanchez of the Naval Postgraduate School. Their assistance in model structuring, Java code development, and model analysis was invaluable.

Additionally, the author would like to thank Dr. Tom Edmunds, and Dr. Brendan Dooher of Lawrence Livermore National Laboratories for their guidance and assistance in formulating this project.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

An exploration of the potential effectiveness of vehicle mounted mobile WMD detectors is conducted via simulation. A simulation model was developed that combines actual roadway data from geo-spatial data sources and discrete event simulation to analyze encounter and detection dynamics in an urban environment. The urban area of study is Washington D.C. and surrounding counties out to a few miles beyond the beltway, encompassing approximately 620 square miles of urban and dense suburban road networks. Various experimental design points are inspected, with specific interest on the return on investment (in the form of probability of detection) depicted as a curve generated by increasing numbers of configured vehicles.

The model is an application of a basic searcher-detector scheme in which many sensor equipped searcher vehicles patrol city streets in an effort to intercept periodic attacks on a high value target area in the center of the city. Sensor vehicles are randomly moving, non-intelligent patrol vehicles without assigned patterns or patrol beats. Penetration by the attacker from outside the area of regard to the outer perimeter of the D.C. beltway is assumed. Attackers, having breached other defensive layers and arrived in the periphery of the city, start attack runs from random points in the vicinity of the beltway and seek a shortest path to the area of the Capitol. Detectors on patrol vehicles are purposefully generic with extremely limited range, and only have the ability to screen vehicles in adjacent, opposite direction traffic on the same road segment. The generic, limited range detectors modeled are similar in capability to field tested nuclear detectors and results generated are transferable to other similar sensor configurations.

The effect of penalizing attacker travel on expressways or highways on their attack run is also explored. This penalization reflects a defensive scheme in which the express routes are overtly patrolled through some other means. In this scheme, the attackers are strongly discouraged from using the expressway as part of an attack route, and are effectively forced “onto city streets,” where they are more readily detected by actual physical devices while in motion.

The intentionally generic sensor modeled generates results that are more accurately labeled as encounter probabilities. The probability of encounter results generated by the model are intended to support a bridge to a more developed physics model of detection and a detailed traffic model capable of arbitrating the intercept of a detected vehicle. To support further exploration of detection and pursuit/interdiction, other measures of effectiveness are presented, including the distribution of intercepts by road speed category and the distance and time remaining on an attack run at time of intercept.

Searcher allocations of less than 50 produce less than a 5% probability of encounter, while allocations of 400 searchers produces an approximate 20% probability of an encounter. Although small, these statistics do represent an appreciable barrier and credible deterrent within a layered defense model against singular, extraordinary attacks. Mobile detectors additionally provide focused screening power inside territorial borders, can augment fixed screening processes in the event of an escalated threat, and provide flexible response options.

I. INTRODUCTION

A. THE CHALLENGES OF WMD THREATS

The coupling of advanced technology, extremist ideologies, and global interconnectivity have presented a tremendous challenge to open societies and their populations in the form of Weapons of Mass Destruction (WMD) threats. Since the dawn of the nuclear era, weaponry capable of producing damage on an unthinkable scale has advanced and proliferated in many areas of the world. It is widely accepted that it is more likely that free societies will face an increased threat from these weapons as political and social tensions embolden our enemies. The magnitude of destruction that these weapons bring in possible loss of human life, destruction to the fabric of our economy, and their potential to alter the course of the nation require extraordinary means to prevent their development, transport, and use.

This thesis is principally an exploration of one aspect of that defensive structure. The overarching goal supported is the hardening of homeland defenses and furthering of the exploration of smart and agile structures which decrease the probability that defensive layers will be penetrated by a willing and dedicated attacker.

Threat capabilities are truly large. Devastation of even a one or two square mile section out of an urban area could have a human cost measured in the hundreds of thousands. Destructive capabilities of this magnitude must be met with extraordinary effort and will require smart, yet extensive devotion of material, personnel, and research effort. Contrarily, budgets for defense are not unlimited and must be wisely apportioned to counter numerous potential threats over vast geospatial environments. Military offensive operations can be focused on generation sources of chemical, biological, or nuclear weaponry if the target can be identified and “fixed” to a location. The effort to extend offensive operations to eliminate these threats at the source has obvious limitations in the forms of accurate intelligence and pervasive battle-space access culminating in the ability to hit a fixed a target. Therefore, offensive operations will never suffice as an effective security measure in and of themselves. Defensive structuring is arguably more difficult, most notably due to the fact that there is really no margin for error. Additionally, the diversity of threats and their compact size present additional problems. Finally, the sheer volume of goods and materiel transported through the global trade network by air, sea, rail and truck make screening of cargo for small, lethal devices particularly challenging. The clear challenge is to match *and exceed* the offensive ingenuity of prospective terror agents with proactive, clever security measures in a cost efficient manner.

B. LAYERED SECURITY APPROACH

Structuring defensive measures in an open, actively trading society is a daunting challenge. The idea of enforcing an "impenetrable security barrier" is an attractive paradigm, but fails on further exploration. Securing even one mode of entry with a near 100% screening process is untenable. Consider, for example, maritime container transport. Within this network, we have almost certain control over the *location* of the points of entry—the same cannot be easily stated for vehicle or airborne traffic—at our numerous major shipping ports, but the challenge of matching the *volume* of traffic with a 100% screening effort is unachievable. Consider, for example, an exemplar of the newer class of container ships, the Emma Maersk, which is capable of delivering 11,000 standard 20-foot container shipments in a single visit (Maersk Line 2007). Erecting screening facilities to match this capacity with pervasive scanning and detection mechanisms is currently infeasible without extracting a high economic cost on transportation and economic infrastructure. Air and vehicular points of entry are faced with similar challenges which are exacerbated by the more numerous points of debarkation, yet are clearly not as challenged in volumetric terms.

A layered approach to providing security beyond points of entry is much more tenable. The addition of random screening or randomly patrolling agents is an attractive addition to any fixed pattern of defense. An attacker planning a singular attack, such as a WMD detonation, will likely have a plan to circumvent static, overt screening methods. Random patrol, unannounced roadside screening, and other similar tactics are much more difficult to defeat through detailed attack planning, and present a distinct advantage to the defender. The defensive power of the strategy of employing a layered approach to securing borders is bolstered by the power of compounding independent probabilistic events, and allows much more flexibility of implementation. For reasons outlined previously, no layer in the defensive structure would be designed for a 100% screening effort, but rather with the intent to make a substantial contribution to a progressive defensive model that in total, provides a very high confidence defensive structure. This structure should (and currently does) have an overseas component in the form of point-of-embarkation screening, a point of debarkation screening effort, and should be augmented with smart and flexible deterrence and screening efforts within the United States and other concerned nations. A prospective attacker would need to penetrate each layer of security independently, and the cumulative probability of success diminishes greatly with a well-layered structure. This structure is depicted in Figure 1. Both security structures depicted, the single layer and the multi-layer, have the same probability of defeating an attack. The multi-layer approach states that a threat device has a 0.30 probability of being detected at embarkation point, a 0.40 probability of being detected at debarkation, etc. Despite the low likelihood of detection at each stage, an attack would need to successfully navigate all wickets in order to execute.

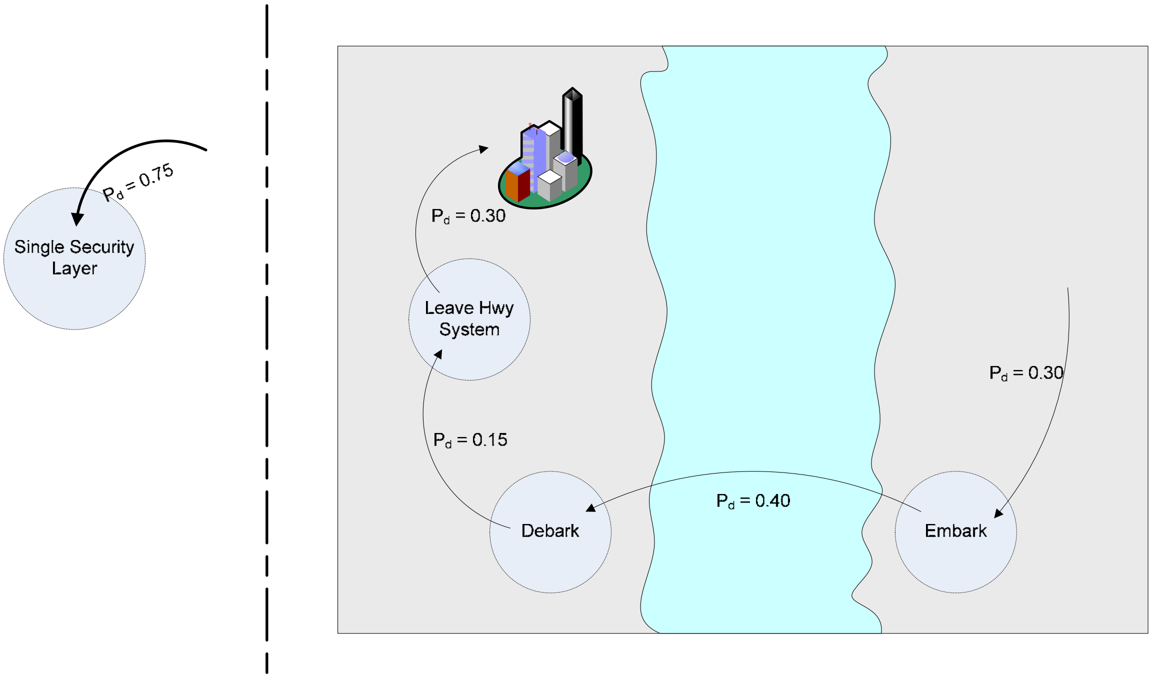


Figure 1. Single Layer Security vs. Multi-Layer Approach

Layering the approach to security has numerous additional merits. First, it is not inconceivable that a WMD threat could be manufactured, stolen, or assembled from innocuous sub-assemblies within the United States. An over-investment in border security would empower the attacker in this scenario. Having already penetrated the single defensive layer at the border, he would enjoy extreme freedom of movement. Structuring defensive capabilities within our borders to provide some layer of screening and search is required to address such scenarios. Additionally, large, fixed, highly structured security measures are poorly matched against patient, smart adversaries. Borders are routinely tunneled under, fixed patrol patterns are observed and defeated, and trial events are staged to probe vulnerabilities. An attacker who has one opportunity to stage an entry into a target area with WMD will make a concerted effort to defeat standing security measures. A layered approach, augmented with a mobile detection element, employs forces and screening devices that can inject a degree of unpredictability, which can be a force multiplier against singular attempts at entry. For example, a roving sentry who randomly reverses course along a large fence line will certainly not provide a guarantee of success against an overpowering attack force, but can provide a very formidable challenge to a small element seeking covert entry. Contrarily, if the sentry routinely walks the entire fence line in a predictable fashion, his observable mode of operation is easily defeated by a patient adversary.

In summary, with respect to screening models, it is clear that an investment in a single "impenetrable" layer at points of entry is not the optimal approach. Although fixed location screening sites and patterns of surveillance are certainly necessary, significant performance is gained by augmenting with mobile or deployable elements and further enhanced with stochastic screening methods wherever possible.

C. MOBILE DETECTION

The ability to screen for WMD threats from a mobile platform is an excellent augmentation to a layered security approach. Mobile units have multiple utilization modes and can make a sizeable contribution to an overall security umbrella. They can be deployed after an intelligence lead on a specific location to form a screening perimeter capable of scanning inbound and outbound traffic to the quarantined zone, augment fixed security screening facilities on a road network during high-demand screening efforts, or they can perform independent screening operations in likely threat locations. This thesis focuses on the potential benefit of an investment in mobile screening technology—specifically nuclear threat detection by vehicle-born detectors. Such detectors are currently operational and have credible detection capabilities against other moving vehicles. The conclusions of this thesis are not constrained to nuclear threat detection studies. This model can be applied to a similar screening layer for many other types of threats. The conclusions are applicable to any mobile detector scheme as long as key underlying assumptions on detector performance match.

For the purposes of the study, a major metropolitan area is selected along with adjacent counties that encompass a geographic subset of the roadway system appropriate in scale for WMD attack study. The city at the heart of the map is provided the defensive capability presented by the mobile detectors. As previously stated, these detectors have multiple employment modes, but only the contribution gained in random patrol mobile screening is presently considered. The scheme of employment is to configure a fixed number of generic patrol vehicles with detection devices and allow them to screen road traffic while either parked or driving within the roadway structure contained in the map. The attacking agent in the model is a vehicle equipped with a generic threat device.

This thesis focuses on the dynamics of individual attack runs by the attacking agents and their potential interactions within networks of deployed mobile sensors. In this baseline study, the attack runs commence at a fixed distance away from the target. Attacks proceed along a shortest path route to the target, where shortest path is determined by driving time without consideration for traffic. Searchers in the model have no pre-knowledge of the attack in progress or its origin. Their movement is random within the geographic area of regard (AOR) in the model.

D. PROBLEM DEFINITION

The area of exploration of this model is to define the effectiveness of this network of mobile detectors. Specifically, the aim is to develop a performance model that can be used to help define a cost-benefit relationship for an investment in this type of security mechanism. Can 50 vehicle-mounted sensors provide a credible layer of protection or deterrence against a vehicle born nuclear threat? Can 200? Clearly, the probability of encounter and detection are inextricably linked to the underlying road structure, capabilities of the sensor, and nature of the threat device. An exploration of the physics of nuclear detectors and sensing is beyond the scope and classification level of this paper, but a thorough investigation into the encounter dynamics is presented. An “encounter” is defined to be an interaction between a searcher and the attacker in which a detection is possible. The encounter dynamics are the necessary sub-structure to support a more detailed investigation of detection declaration and interdiction.

Data specific to the nature of the encounter will be of critical importance to determining the likelihood of a credible detection. A generic nuclear detector may have sensing capabilities against various threats out to a range of only a few meters and require a short, yet non-negligible dwell time for sensing to occur. Therefore, characteristics of the road network such as speed limits and lane counts (and resultant road width) are required to step from an encounter model into a detection model. Dwell time can be extrapolated from assumptions about closest point of approach (CPA) and closure velocity between two vehicles moving in opposite directions. Lane count in two directions will underscore a probabilistic model for the range of the encounter. A detector with a theoretical 2-meter detection capability may not be likely to detect a threat on an urban street across four lanes of traffic, but would most certainly be effective if the encounter occurred on a 2-lane road.

E. PREVIOUS APPROACHES TO MODELING

The area of exploration in the model is a random search approach to detecting a mobile attacker. Various random search algorithms exist for open fields of movement and are widely studied and used in anti-submarine warfare and ocean surface search (Wagner et al. 1999, 181). The underlying cumulative probability of detection formulation in these models has the functional form

$$F_d(t) = 1 - e^{kt}$$

where $F_d(t)$ is the cumulative probability of detection and k is a function of sensor scan width and velocity. This model is difficult to justify in an urban environment because of the constrained set of movements (a network of roads, bridges, tunnels and highways) and the attacker’s traversal through the network during the search.

A more applicable approach utilizes Markov chain analysis of the AOR. This approach discretizes potential locations within the road network and analyzes movement between nodes as Markov state transitions (Edmunds 1994). This approach has the advantage of representing attacker movement *through* the AOR and is a more accurate representation of searcher movement. Searcher direction changes occur as state transitions between states in the Markov matrix, which more accurately represents turn choices at intersections.

The principal shortfall of this approach is the challenge of dimensioning of the Markov matrix to properly represent the search grid and the discrete steps the attacker and searchers take. Edmunds takes an approach of this type in analyzing attacker penetration into the center of an urban area in an initial analysis of potential benefits of mobile detectors in the Washington D. C. area (Edmunds et al. 2006). Although this effort advances an understanding of the dynamics of the city street search problem, it makes assumptions as to the construction of the search grid and physical relationships inherent to the actual road network. It is well suited for grid-like city constructs, but fails to adequately model expressways, circuitous street constructs, and suburban road structuring in general.

F. INTRODUCTION OF THE ROAD NETWORK MODEL

A review of prior work suggests that an accounting of actual road networks generates better detection models. Dense urban grids, bridges, expressways, and encircled subdivisions challenged by linear road connections suggest that the underlying geometry and traffic flow designs must be accounted for in model construction.

When reviewing an attack model to explore urban penetration, the origin of the attack and the possible detection zones should be considered to determine a suitable radius to circumscribe the AOR. The model developed in this thesis centers on urban defense, and presupposes the transportation of a threat device to within reasonable striking distance of the target. Excluded from the model are suburban areas of extraordinary distance from the attack target and, in general, the interstate highway system connecting the remainder of the country. Bounding the AOR is necessary to contain the motion of the patrol vehicles. The model developed is a discrete event simulation that uses actual road network information.

Chapter II will explore the model utilized and include a discussion of key assumptions and of discrete event simulation in general terms. Chapter III will cover results and observations and Chapter IV will cover possible performance enhancements, a discussion of countermeasures, and possible extensions to the model schema.

II. MODEL DEVELOPMENT

The basic construction of the model is detailed and explored. The model utilizes a unique blend of data describing the actual physical structure of the road network and discrete event simulation. Geospatial data is parsed and transformed into a graph structure. The resultant graph in node-arc format is used as the basis of the motion model in the discrete event simulation. A more detailed description of the implementation is provided as well as a discussion of independent variables and governing assumptions.

A. MERGING OF DISCRETE EVENT SIMULATION AND GEOSPATIAL DATA

The inextricable linkage of the problem statement to the geographically defined characteristics of the AOR suggests that simulation rather than closed-form mathematical analysis is the course to pursue. The underlying structure of an urban road transportation network is uniquely characterized by civic design to include bridges, expressways, one-way routings, divided road structures, and other such civil-engineering constructs that will uniquely determine the probability of encounter between two entities moving within the network. Furthermore, should two entities encounter or sense each other, the outcome of the interaction, whether it be a detection, detection failure, or false detection, will be driven in part by the speed of closure between the two and the separation at time of passage, both of which are determined by the underlying road structure. These complexities are challenging to incorporate into an analytic model. The reliance on actual data for road networks requires the introduction of geospatial data.

The use of geospatial data conveys many benefits. Consider two examples which illustrate some of the core advantages. In an urban area that is dominated by one-way street routings, a moving detector may have very little opportunity to encounter another moving target if both were in the same flow of traffic simply due to the lack of opposing traffic on the streets. However, if the roadways entering the target area were predominately two-way streets, the chances would be much improved. Other geospatial challenges are easily tackled by actual data. A city with significant sprawl encompassing dense suburban neighborhoods will generate different patrol results than would a city with more direct highway access, less dense adjoining suburbs, or one that is geographically constrained by a coastline, for instance.

Discrete event simulation (DES) was selected as the exploratory tool to model movement within the road network and interaction of the entities therein. Briefly, discrete event simulation maps distinct changes in the state of an object into discrete events that happen during the simulation. As a matter of implementation, as an entity enters a particular road segment its *location state* is altered by the *event* of selecting or proceeding on a particular segment. It is under this construct that movement is modeled on the road network.

B. GEOSPATIAL DATA AND CONVERSION TO USEABLE FORMAT

Geospatial data is a mathematical representation of physical structures with geographic specificity. This form of data is widely used in many contexts. For instance digital terrain elevation data feeds numerous flight simulation engines in order to reproduce terrain elevations in virtual worlds. Road and rail networks have been painstakingly converted (by mapping agencies) into geographic coordinate networks in order to support routing programs and interactive mapping tools such as many internet mapping websites, quickest route tools, and GPS backed routing devices. Datasets to support these tools and operations are available both commercially and through websites such as the National Atlas (National Atlas 2007). This thesis utilizes a commercially available dataset (NAVSTREETS v3.3.0) for roads and highways in the continental United States in the widely utilized “shapefile” (.shp) format. This data format separates roadways into individual segments, divided at every intersection, that are tagged with a multitude of descriptive data including speed categorization, direction of travel (applicable to 1-way streets), postal coding, street naming, etc. Similar datasets are routinely used for mapping programs to compute driving directions or produce maps in specified areas of interest.

1. GeoTools Open Source Library

In order to utilize descriptive road data for simulation, conversion into a format suitable for the simulation engine is required. The DES engine in this case is a Java-based program constructed to run on a graph (node, arc) structure. The means to render a useable graph network data structure from raw geospatial data is embedded in numerous commercial routing packages, but is also available in open-source format through various open-source Java libraries. GeoTools provides such methodology and it was selected as the analytical tool to enable this simulation (GeoTools Project, 2006). The GeoTools library provides an array of Java-based implements to analyze, plot, catalog, and manipulate geospatial data from various input formats.

2. Dataset Conversion

A graph-generating Java Class derived from GeoTools converts Shapefile street data into road network graphs in node-arc format. This generating scheme utilizes a graph-building algorithm that converts the raw geo-spatial locating data, in the form of coordinate arrays, to produce a useable node-arc structure divorced from actual geospatial coordinates (latitude-longitude). This process produces a graph that has a node for each intersection in the actual network. The baseline code of this conversion program was substantially modified to enable the creation of directed arcs to reflect actual direction of travel on the arcs produced, which is essential in representing realistic traffic flows. The production of distinct nodes for geospatially distinct artifacts is critical to understanding how the simulation will handle intersections and opposite direction traffic, specifically on divided roads. Road segments labeled as bi-directional that are represented by an array of coordinates will produce two arcs (one for each direction of travel) between the two nodes at opposite ends of the same street segment. However, when roads are divided by a significant median, *and* represented by two distinct coordinate arrays, they are rendered as individual arcs between independent node sets without an opposing edge. This is the norm for divided highways or expressways, and can be implemented in roadways with very wide medians. For purposes of the model, no interaction may occur across such a divide. The potential for interactions exists only on opposing arcs between node sets that are geographically equal on both ends of a street segment. This relationship and the translation of road data into graph data is depicted in Figure 2. The underlying construct in which a sensor has a very limited lateral range curve supports the simulation consequence of excluding interaction across large medians. Considering the geometry of Figure 2, a vehicle traveling west on Main St. would be unable to sense a vehicle traveling east on Main St.. However, vehicles traveling in opposite directions on Tree St. between 2nd and 3rd Avenues could detect each other.

Weighting of edges in the constructed graph is implemented in order to distill travel times and facilitate computations to support attacker shortest path routing. The edges are weighted with a baseline travel time generated by a summation of the geodetic calculations of length of all included segments and dividing by the speed categorization of the road segment.

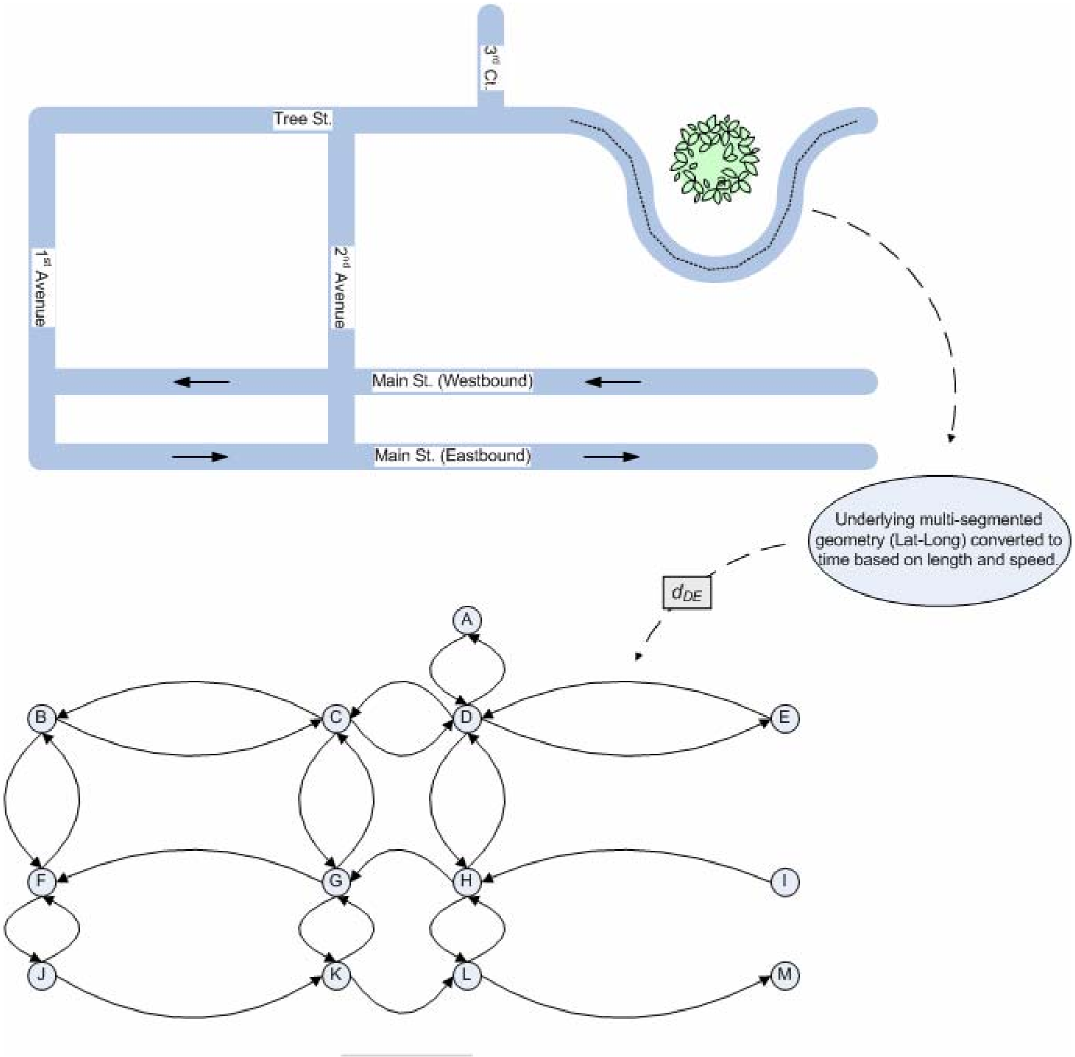


Figure 2. Conversion of shapefile street data into graph (node, arc) format

Trimming the graph to characterize operations within one specific geographic area is a subjective process. As stated previously, an attack route is assumed to start outside the metropolitan area and makes its way to a centrally located node. The tailored area should reflect reasonable inclusion/exclusion decisions on affected patrol areas (districts, counties, police jurisdictions, etc.). In a subjective look at the principal AOR analyzed, the D.C. beltway serves as a reasonable geospatial reference by which to sever the simulation model from the entire U. S. road network. The resulting map is depicted in Figure 3. Potential starting nodes for an attack are generated with a geometric distance filter set at approximately 10 miles from the Capitol area.

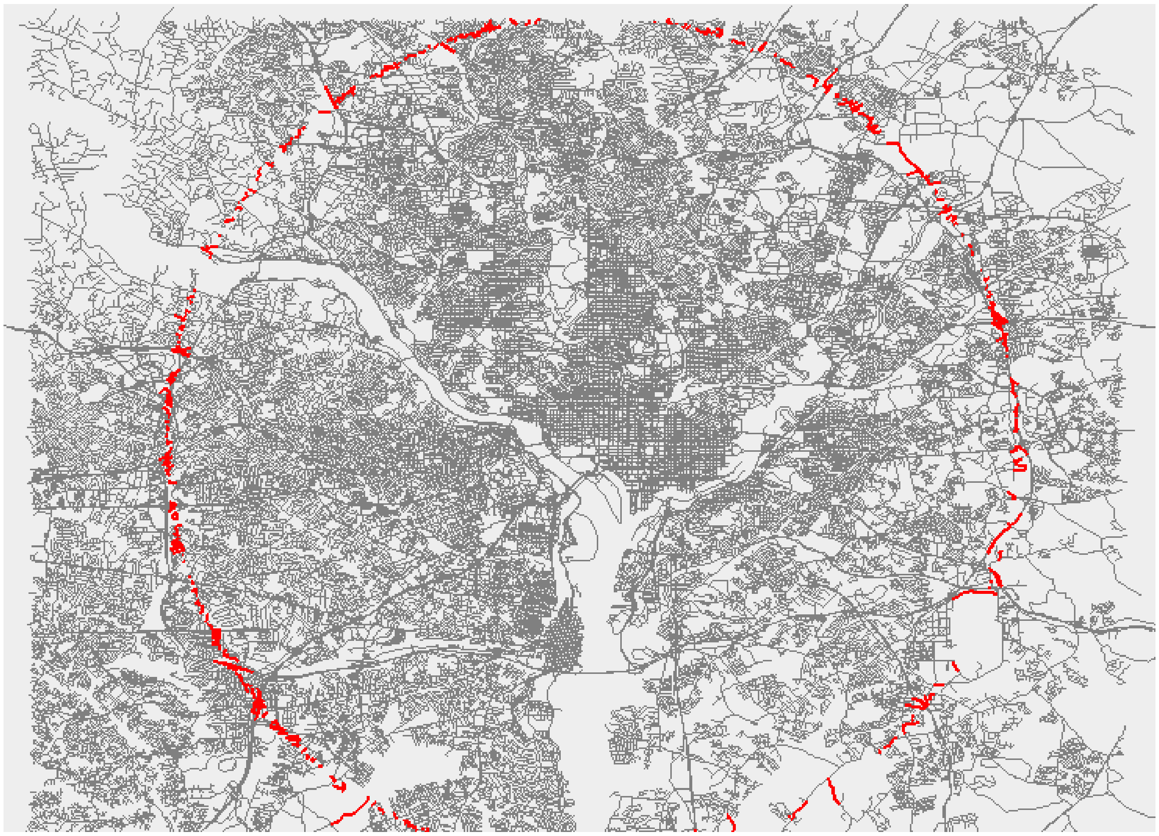


Figure 3. AOR processed by simulation: Washington, D.C.
Red road segments represent 458 distinct possible attack starting locations on a 10 mile radius from the Capitol. The D.C. Beltway (I-495) is encircled and is visible and roughly concurrent with the east and west portions of the attack ring.

C. FUNCTIONAL FLOW OF THE PROGRAM

Prior to running the DES, the dataset is converted and the model is initialized. These steps are enumerated in Table 1.

Stage	Functional Task
1.	Load geospatial dataset. This dataset is a pre-trimmed shapefile (.shp) with a feature set exclusively consisting of roadways in the AOR
2.	Label the “target” node in the network, and a “home base” for searchers to reset to.
3.	Geographically filter the dataset to produce a set of intersected arcs to serve as potential starting nodes for an attack. This is accomplished by a circular intersection filter that gathers any road segment that intersects a circular polygon set to a diameter of 95% of the minimum rectangular dimension of the map.
4.	Convert the geospatial data (roads with arrays of Lat-Long coordinates to describe their position on the Earth) to Graph (node, arc) format. One-way streets are implemented as directed arcs based on annotated direction of travel. Standard two-way streets are entered twice, once with geometry reversed in order to produce opposing arcs between the same two nodes. Weight the edges by travel time developed from a curvilinear length calculation and the speed category of the road.
5.	All graph nodes are screened by reaching and reverse reaching algorithms to develop sets of unreachable (U) and “isolated” (I) nodes, s.t.: U = { nodes cannot be navigated to from the home base } I = { nodes that do not have a path back to home base }
6.	Remove any nodes from the starting location set that are a member of set I. Then, use Dijkstra’s algorithm to generate and cache shortest paths from all potential attack origins to the target node.

Table 1. Functional flow of program before to simulation run.

Handling of special case nodes is extremely important to control undesirable behavior within the simulation. To support categorization of problem nodes, reaching and reverse-reaching algorithms are employed to label nodes within the dataset. Problem nodes are categorized as either unreachable or as isolated. Unreachable nodes are not accessible by any path from the home base. Isolated nodes have no path returning them to the home base. Both are frequently created as a byproduct of trimming an urban subset graph from the larger road network. Odd subdivisions or divided highways can produce multiple problem nodes, depending on location of cut set. For example, in the simple network from Figure 2, if Node B is the base node then Node I is unreachable and Node M is isolated.

Basic statistics for the Washington D. C. Graph are presented in Table 2. It is implied at initialization that the home base node and target node are strongly connected, i.e., that each can be reached from the other. Handling the inadvertent placement on or entry to an unreachable or isolated node is discussed in the motion control of each entity.

Graph Characteristic	Set Size
Directed Edges	200,487
Unreachable Nodes	306
Isolated Nodes	301
Attack start nodes at 10mi radius	458

Table 2. Graph Statistics: Washington D. C.

D. DISCRETE EVENT SIMULATION FUNCTIONALITY

1. Core Movement and Movement Management

The DES phase of the model is run subsequent to the preparation steps described in Table 1. The functional flow of the simulation is more thoroughly detailed in the Event Graphs which follow.

The DES is conceived and run within SimKit, a Java-based discrete event simulation development library (SimKit, 2007). Within SimKit distinct simulation entities are embedded with process logic, and are capable of scheduling events on a master schedule. Entities within the simulation are capable of interacting with other entities by triggering events for each other through a "Sim Event listener" pattern supported within SimKit (Buss and Sanchez, 2002).

The basic construct of the simulation is that an ***entity*** (either a searcher or attacker) is assigned to a ***mover*** that is capable of traversing edges in the graph created. The mover's actions are dictated by a ***manager*** that assigns its next move based on management type (by path or random motion). Managers get their instruction from a controlling entity; which is an ***attack instigator,*** or in the case of the search team a ***search leader.*** The basic schema for both a mover and a manager are reproduced in Figures 4 and 5.

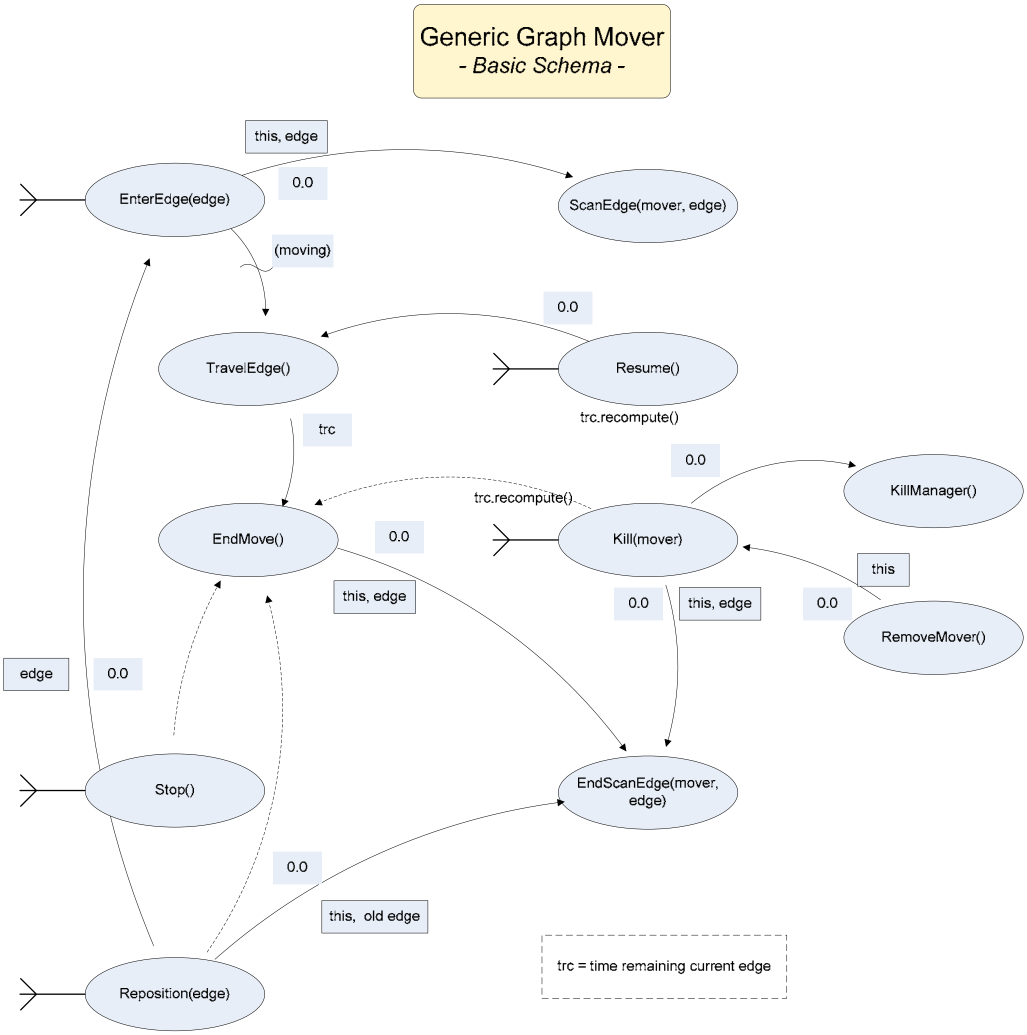


Figure 4. Graph Mover: Basic Schema for both searcher and attacker entities.

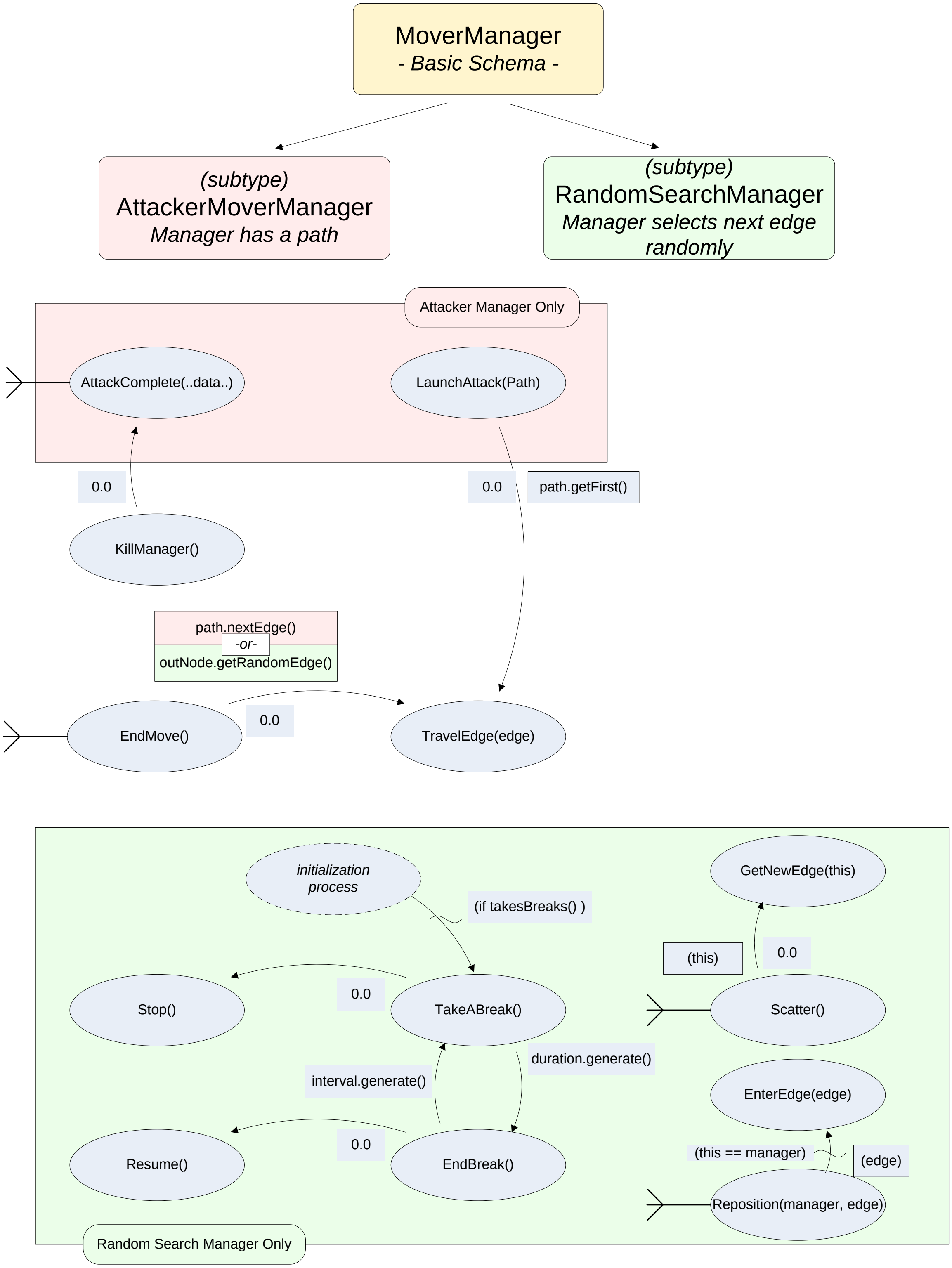


Figure 5. Mover Managers: Basic Schema largely shared. Differences between searcher and attacker are as noted

2. Sensing and Arbitration

A ***mediator*** element is instantiated to adjudicate detections between the moving elements and arbitrate interactions. The Mediator event graph is depicted in Figure 6. The mediator monitors the positions of all searchers and of the attacker. Each time any mover element enters a new edge on the graph, its position is updated and all edges in opposite direction between the same two nodes (there should be no more than one) are inventoried for the opposite type entity. If an attacker is present on an opposite direction edge when a searcher enters an edge, an arbitration is scheduled. If, by chance, multiple searchers are present, each interaction is arbitrated until a kill is declared.

In the current model a perfect sensor is assumed, therefore every encounter within detection radius is declared a detection, and nominally an interdiction. Thus, the model as developed is an encounter model. All encounters are currently detections and are declared kills administratively. The physics model necessary to convert closure velocity and lane separation into dwell time, sensing distance, and subsequent detection rates over numerous threat types would bring this paper into higher security classification levels, and is therefore left as a suggested extension. Another alternative would be to model detection and interdiction using probability distributions in place of detailed physics models. In either case, the core factors that would empower a more complete detection model are driven by the results provided in this thesis.

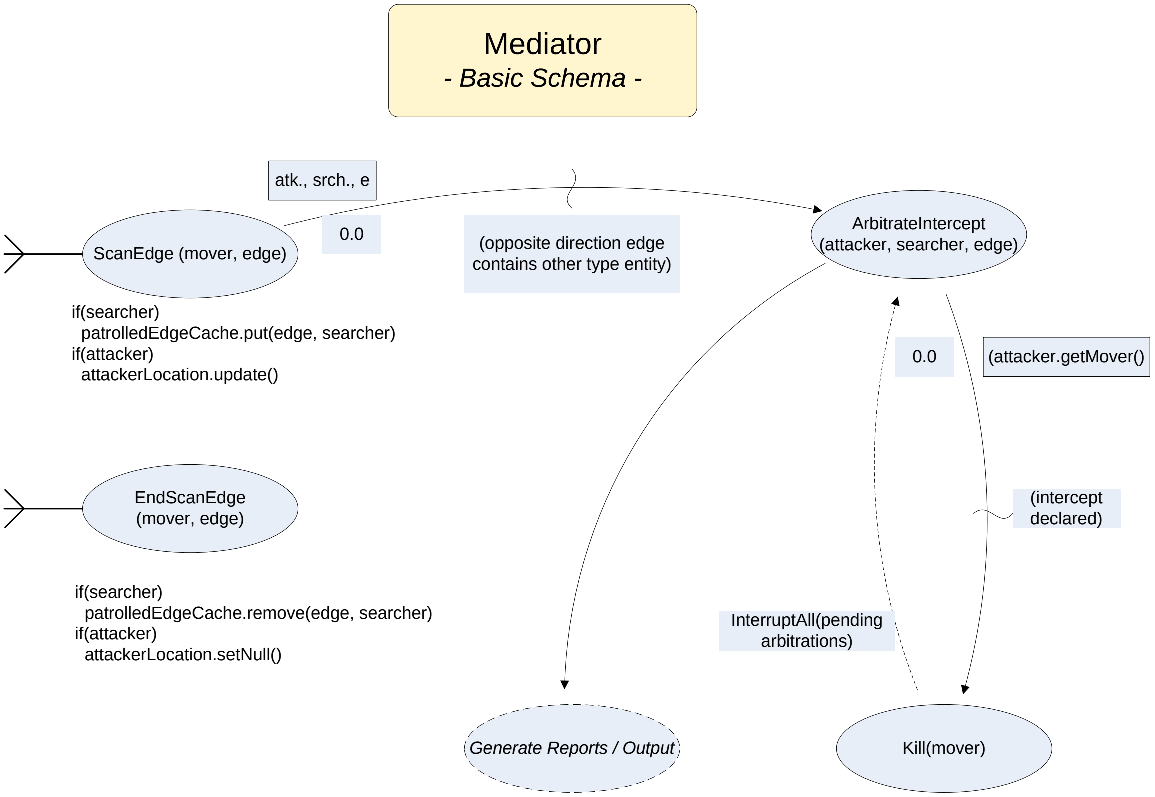

Mediator
- Basic Schema -
atk., srch., e
0.0
(opposite direction edge contains other type entity)
ScanEdge (mover, edge)
ArbitrateIntercept (attacker, searcher, edge)
if(searcher)
patrolledEdgeCache.put(edge, searcher)
if(attacker)
attackerLocation.update()
0.0
(attacker.getMover()
EndScanEdge (mover, edge)
(intercept declared)
if(searcher)
patrolledEdgeCache.remove(edge, searcher)
if(attacker)
attackerLocation.setNull()
InterruptAll(pending arbitrations)
Generate Reports / Output
Kill(mover)

Figure 6. Mediator: Basic Schema and functionality

3. Assembling the Model with Listeners

As stated, the model is connected via Sim Event listener patterns. Upper level entities direct the action of lower level movers and receive feedback should an intercept occur. The number of searchers is determined by instantiation routines and can be varied and supported by the one-to-many relationship depicted in Figure 7.

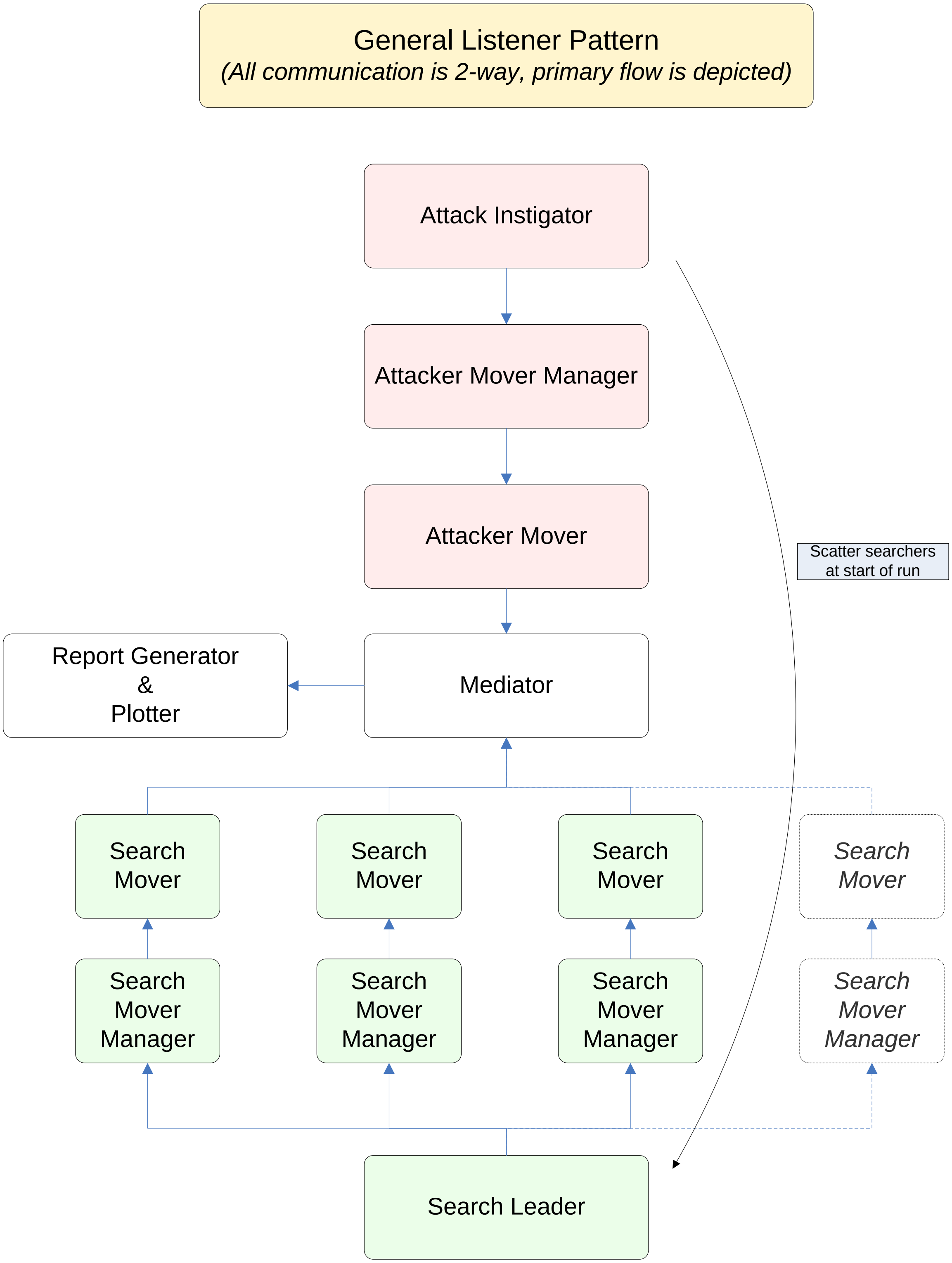


Figure 7. Listener pattern setup and interaction map.

Each attack is an independent stochastic replication. Sources of randomness in the run are:

- Searcher location prior to start of run. All edges in the graph are enumerated, and all searchers are randomly re-assigned prior to the commencement of each run.
- Random selection of the attack start node from among the eligible start set.
- Searcher movement on the map driven by random outbound segment selection at each intersection.
- Searcher break-taking interval and duration is randomized by a uniform random variable with reasonable limits.

4. Searcher Movement

Searchers are initialized at the beginning of every attack run. The initialization process uses a random number generator to assign each searcher to a random starting arc within the AOR. Specialized handling after initialization is only required if the searcher wanders into an isolated area (characterized by nodes that have no return routing to the home base). In this instance, the searcher is reset to the home base. This results in a very minor increase in searcher density near home base, which is reasonable, if not an underestimation of patrol effort near the "headquarters." The home base currently employed in the Washington, D.C. model is a few miles north of the Capitol, which is the target.

Searcher movement within the network is random after an attack run commences. No pre-defined routing or intelligent patrol algorithm is implemented. At each node (intersection), the searcher simply "rolls the dice" to determine which segment to enter. Currently, a no U-turn algorithm is employed. The no U-turn algorithm precludes the searcher from selecting an arc that would return him to his previous departure node unless it is the only option available (as in the case of a dead-end street.) While a real searcher vehicle may employ U-turns occasionally, it is postulated that no U-turns is a much more realistic assumption that will: (a) provide more forward motion to search components, (b) result in a searcher with less “jitter” in his routing, (c) prevent a completely random searcher from making a U-turn probabilistically every four blocks on average, assuming four-way intersections.

Searchers periodically take breaks, as all patrolling entities are apt to do. Scanning by the searcher during break continues on the arc that he commences his break on. Currently, the scan only includes one direction of travel, exactly as if he were in motion. This is, again, a reasonable extension of the pretense of a limited range scanning device that may only be able to “see” into one adjacent lane. Currently breaks occur for each searcher independently on a uniform random interval [0.2, 1.0] hrs. The duration of a break, once it occurs is scheduled for a duration by a random uniform variable between [0.5, 1.0] hrs.

5. Attacker Movement

Attackers are instantiated within the simulation and are given a Dijkstra-generated shortest path route to the target. They proceed from node to node along each arc of the assigned path to eventually arrive at the target, unless they are detected and removed. All entities within the simulation travel “at the speed limit” annotated by the speed category for their present location based on the assumption that they would not wish to attract the attention of local police. Traffic conditions are not factored in, nor are traffic flow control devices such as stop signs and stoplights. It is assumed that these devices would have equal effect on all movers within the structure, and their effect should be marginal.

An extension to the attacker shortest algorithm is coded to explore an expanded sample space. A “penalized edge weighter” allows the Dijkstra algorithm to excessively penalize roads with speed categorizations above a fixed value. This penalized weighter is used to explore an alternate solution space in which attackers would never prefer to take an expressway routing to their target. The driving thought here is a proposal that highways may be screened by other fixed or re-deployable screening systems overtly in an effort to screen higher density traffic flows more thoroughly and deter their use by an attacker. This would force any would-be attacker onto the “city streets.” This employment strategy would force the attacker into a more vulnerable position with respect to mobile detectors with limited sensing radii and a preclusion for slower closing velocities. In the simulation, the majority of the highways and expressways that would be subject to penalization are geospatially coded as two opposing one-way highway segments due to excessive lane separation. Resultantly, if they were not penalized, a detector who opposes an attacker on a highway segment would be *unable* to detect him on these divided, high-speed roads.

A restatement of the implications of the penalized edge weighting system is as follows: If the penalized weighting is NOT used, many attack routes will include lengthy segments of interstate or highway use by the attacker. He will be essentially undetectable on the majority of these segments within the simulation due to geographic lane separation. This is compatible with assumptions of detector design. If the penalized weighter IS utilized, attackers will avoid the highway segments unless absolutely necessary (and if needed, will travel them at normal speed). Searchers are unaffected by the penalized weighting and select all moves randomly.

In summary, the model progresses in two distinct phases. First, the dataset is processed from geo-specific locating data in the form of lat-long coordinates into a graph of node-arc format. Additional administrative steps classify problem nodes, develop a starting node set, generate attack paths, and prepare the network for simulation use. The DES engine then replicates numerous stochastic, independent attack replications using the graph and gathers statistical results. A tiered pattern of Sim Event listeners enables a hierarchical control structure over the entities, and a mediator element monitors both attacker and searcher movements for interactions.

Chapter III relates the independent variables to the control factors discussed. Additionally, the principal results are presented along with a detailed exploration of the distributions of factors which would govern the performance of fully developed interdiction model.

THIS PAGE INTENTIONALLY LEFT BLANK

III. RESULTS AND ANALYSIS

The sample space of the model is parameterized by three factors: the number of searchers configured; penalization of highway usage by attackers; and whether the searchers take periodic breaks or are in continuous motion. Raw results across all configurations are presented along with a multiple logistic regression model and further analysis of the dynamics of the encounters catalogued over multiple runs.

A. DESIGN OF EXPERIMENTS

1. Underlying Goals

The underlying reasoning for using a simulation on actual road networks was to capture the effects unique to a given large-scale roadway system. In this regard the principal independent variable selection was the choice of city for the comparative study. Washington, D.C. was selected first and foremost because it is an obvious high value target that warrants additional protection measures and a layered defensive structure, but also because of its dynamic urban and suburban roadway structure that is characterized by rivers, bridges, express routes into the center of the city, and a well defined encompassing feature, the D.C. beltway.

The principal question under investigation is a return on investment question for a procurement of mobile detectors. Setting aside the engineering considerations and physics of detection, the core question is: What probability of detection does a given number of mobile detectors yield? Obvious follow-on questions of employment tactics, positioning, counter-countermeasures, and what to do when the “red light on the dashboard illuminates” quickly arise, but are all secondary to the need to know how often a detector-equipped vehicle might encounter an adversary.

2. Factors

The primary factor under evaluation is the **number of searchers** invoked. Searcher count is varied from over a range of 50 to 3000. Fifty units represents a reasonable first investment in the technology, and although 3000 is well outside any expected appropriation for costing reasons, it is included simply to confirm simulation performance. The principal range of investigation is 50 to 800 searchers.

Additionally, the effect of the searchers' **breaks** at random intervals throughout the simulation is investigated to determine if the additional motion of the sensor package within the network produced by disabling breaks results in any incremental improvement of detection capability. Recall that a searcher on break is simply motionless (parked) but retains the same scanning ability as a searcher in motion. This may not fully capture true break dynamics in which the search vehicle may be in a parking lot and outside of sensing range of nearby threats (pessimistic scenario) or a searcher who positions his car on or near a median and is capable of sensing traffic in both directions (optimistic scenario); but should capture any benefit added by simple motion of the sensor.

The third factor under direct inspection is a variable controlling the **speed penalty** that is aimed at forcing the attacker off the expressways and onto city streets. When the penalty is enabled, roads with speed limit categories above a pre-defined threshold become grossly penalized when considered by the Dijkstra shortest path algorithm that specifies attack routes.

B. PRINCIPLE RESULTS

The probability of detection plotted against searcher count is summarized in Figure 8. The area encompassed in the AOR represents approximately 620 square miles of urban and suburban development including many Virginia and Maryland counties.

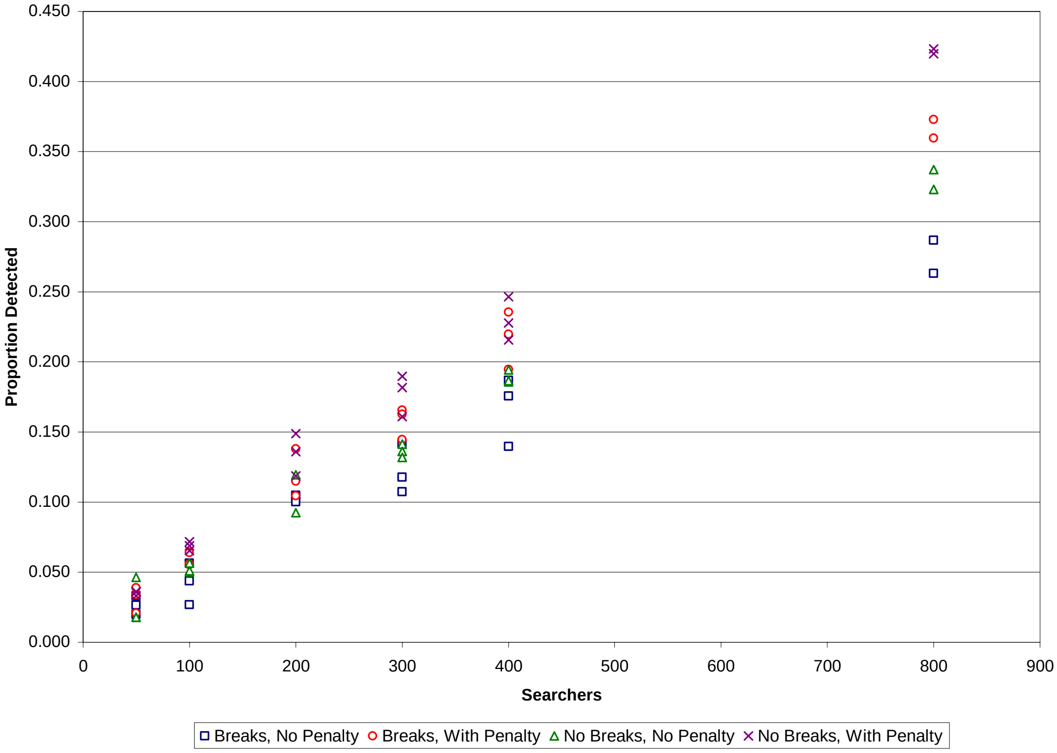


Figure 8. Detected Proportion vs. Number Searchers under various design settings

Each data point represented in Figure 8 represents a 200 (simulated) hour run, which produces approximately 450 attacks with varying time duration of approximately 20 minutes. Design set points for each configuration of searchers, break taking (T/F), and penalized highway routing (T/F) were replicated three times with different random number seeding. At the extreme level of sensor employment, multiple 3000-sensor runs were executed in two basic configurations as a validation of shaping assumptions. The raw data suggests a nearly linear relationship between probability of detection and number of searchers employed over the small band of detection probabilities generated. It also suggests that penalizing the highways makes an increasingly large contribution to probability of detection over the span of number of searchers employed.

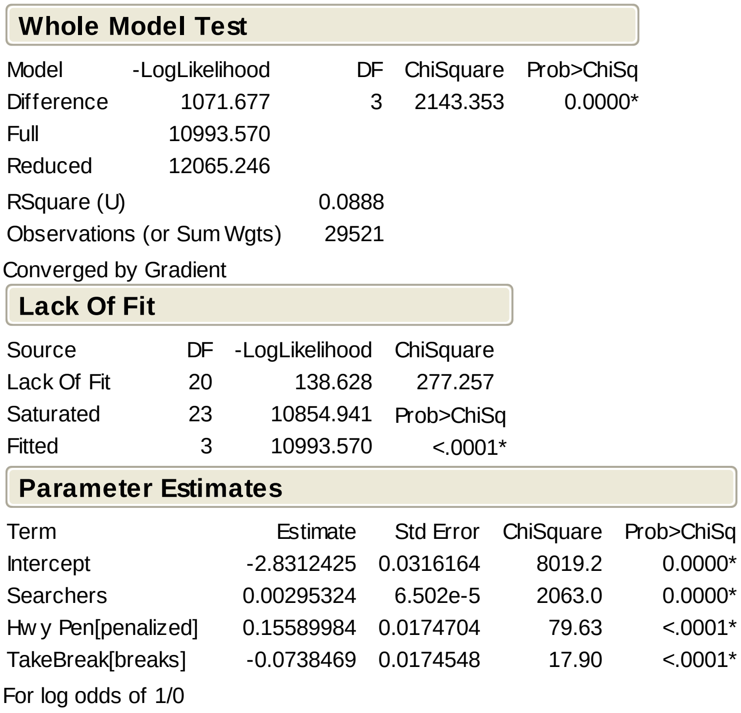
Disallowing searcher breaks yields a slight, yet systematic increase in probability of intercept. Data from comparable runs with 300 searchers is explored in Table 3. These data show an increase of 1.5% in probability of intercept when break-taking is disabled and searchers have more resultant motion in the graph. The subsequent logistic model reaffirms that break taking is a statistically significant factor in the overall logistic regression model, but that its contribution is marginal.

Searchers	Highways Penalized	Take Breaks	Intercepts	Attacks Attempted	
300	FALSE	TRUE	50	465	
300	FALSE	TRUE	67	474	
300	FALSE	TRUE	55	467	
			172	1406	***12.2%***
300	FALSE	FALSE	67	474	
300	FALSE	FALSE	65	477	
300	FALSE	FALSE	62	470	
			194	1421	***13.7%***

Table 3. Marginal Effect of Disabling Break-Taking.

C. FITTING A LOGISTIC MODEL

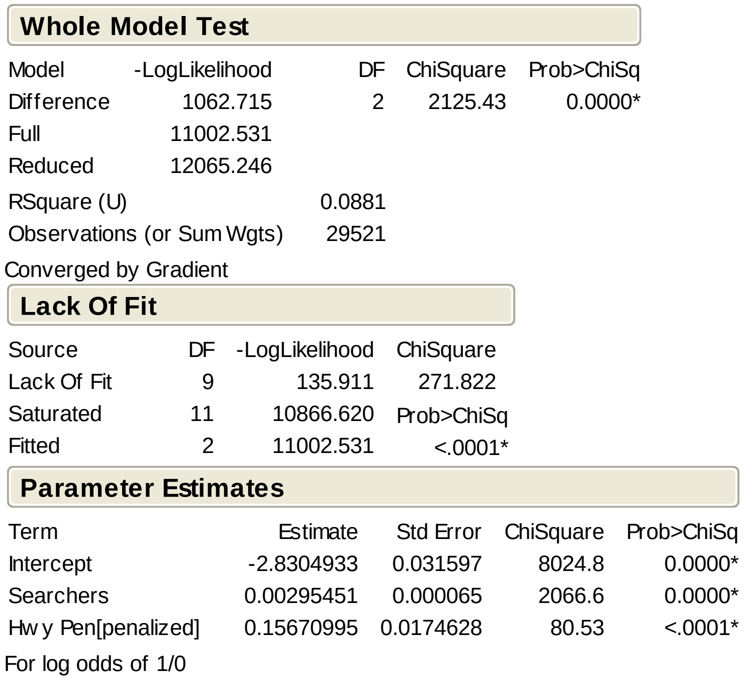
A logistic regression was fit to the model data to formalize the relationship of the three predictor variables; specifically, the number of searchers, penalization of the highways for attack runs, and whether the searchers take breaks. Data used in the regression model encompassed searcher numbers from 50 to 800, which represents possible employment configurations, and encompasses over 29,500 simulated attack runs. Results are depicted in Figure 9. Not surprisingly, the nearly linear relationship of detections to searcher count which spans this lower end of the probability of detection curve gives rise to an exceptional fit, which is significant above the 0.99 level.



```
Whole Model Test
Model        -LogLikelihood        DF   ChiSquare   Prob>ChiSq
Difference        1071.677          3    2143.353      0.0000*
Full             10993.570
Reduced          12065.246
RSquare (U)                    0.0888
Observations (or Sum Wgts)      29521
Converged by Gradient
Lack Of Fit
Source          DF  -LogLikelihood  ChiSquare
Lack Of Fit     20         138.628    277.257
Saturated       23       10854.941  Prob>ChiSq
Fitted           3       10993.570     <.0001*
Parameter Estimates
Term                     Estimate   Std Error   ChiSquare   Prob>ChiSq
Intercept              -2.8312425   0.0316164      8019.2      0.0000*
Searchers              0.00295324    6.502e-5      2063.0      0.0000*
Hwy Pen[penalized]     0.15589984   0.0174704       79.63      <.0001*
TakeBreak[breaks]      -0.0738469   0.0174548       17.90      <.0001*
For log odds of 1/0
```

Figure 9. Logistic Regression with 3 factors: P_d~(Number Searchers, Highway Penalization, Searcher Break Taking)

Although break-taking as a factor is statistically significant, the reduced model without break-taking as a factor still is significant at a 99% confidence level, and is presented in Figure 10. Using the logistic formula generated by the full logistic regression, prediction curves are generated in Figures 11 and 12. In both formulations, the binary regressors are represented by one in the positive state, and by zero in the false state.



```
Whole Model Test
Model        -LogLikelihood          DF   ChiSquare   Prob>ChiSq
Difference        1062.715            2     2125.43      0.0000*
Full             11002.531
Reduced          12065.246
RSquare (U)                      0.0881
Observations (or Sum Wgts)        29521
Converged by Gradient
Lack Of Fit
Source              DF   -LogLikelihood   ChiSquare
Lack Of Fit          9          135.911     271.822
Saturated           11        10866.620   Prob>ChiSq
Fitted               2        11002.531     <.0001*
Parameter Estimates
Term                     Estimate    Std Error   ChiSquare   Prob>ChiSq
Intercept              -2.8304933     0.031597      8024.8      0.0000*
Searchers              0.00295451     0.000065      2066.6      0.0000*
Hwy Pen[penalized]     0.15670995    0.0174628       80.53      <.0001*
For log odds of 1/0
```

Figure 10. Reduced Logistic Regression Model P_d~(Number Searchers, Highway Penalization)

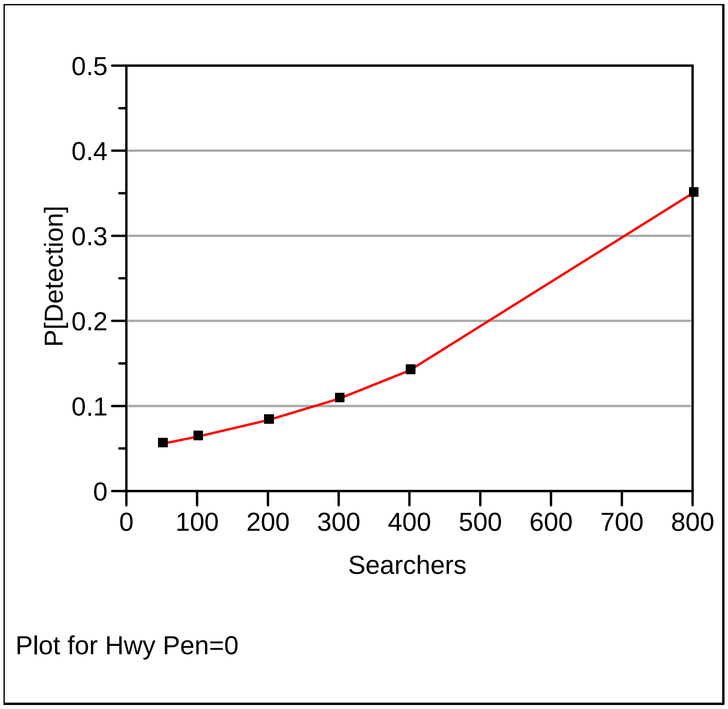


Figure 11. Prediction curve for P_d as a function of Searchers, with highways NOT penalized for attackers

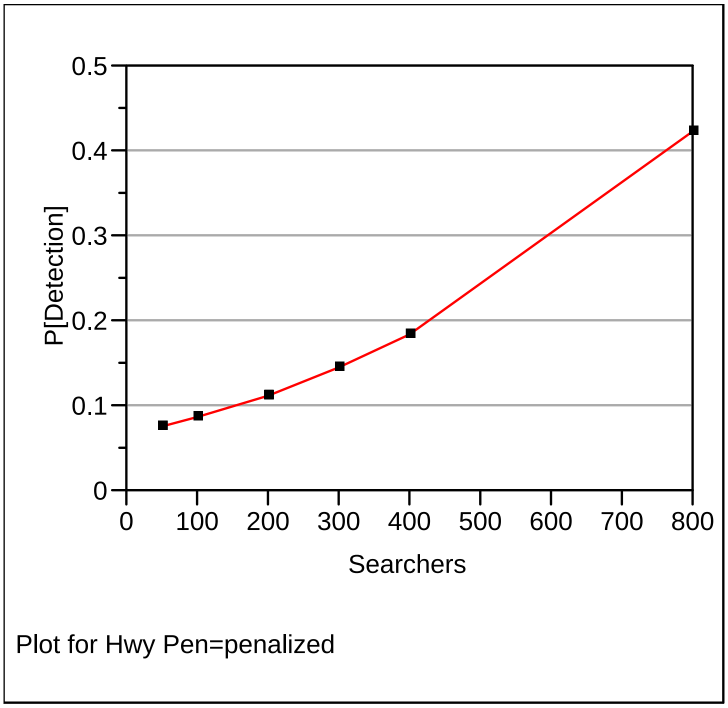


Figure 12. Prediction curve for P_d as a function of Searchers, with highways penalized for attackers.

The nearly linear relationship between searchers employed and probability of detection over the observed range is not surprising, as the number of searchers employed has a nearly direct relationship with the number of street segments patrolled within these parameter ranges. The number of potential searcher locations is the size of the edge set, 200,487. This is more than 250 times larger than the number of searchers employed at the highest setting, 800. In configurations with this disparity in the magnitude of the ratio of searchers to searcher locations, nearly every searcher added directly translates into additional road segments screened with little duplication of effort.

D. EFFECTS OF PENALIZING HIGHWAYS FOR ATTACKERS

Highway penalization in attacker route selection produces systematically higher probabilities of intercept. Again, exploring a characteristic configuration of 300 searchers, a significant increase in intercept probability is noted in Table 4. It must be re-emphasized that this increase in probability of intercept is uniquely characteristic to the highway structure approaching the center of the selected target area; in this case, Washington, D.C. It cannot be extrapolated to other urban target areas without a comparison of highway structures inside the attack radius which may provide rapid transit to the area of the target. Consider, for example, a metropolitan area without expressway routes from the attack periphery into the target area. In this scenario, highway penalization would be completely ineffectual, as there would be no highway segments on any of the shortest routes before penalization. In the simulation conducted, highways labeled at or faster than Category 3 were penalized, and there are numerous expressways penetrating from the beltway inward that are effected. [Category 3 in the NAVSTREETS database is 55-64 m.p.h.].

Searchers	Highways Penalized	Take Breaks	Intercepts	Attacks Attempted	
300	TRUE	TRUE	59	408	
300	TRUE	TRUE	68	411	
300	TRUE	TRUE	67	412	
			194	1231	***15.8%***
300	FALSE	TRUE	50	465	
300	FALSE	TRUE	67	474	
300	FALSE	TRUE	55	467	
			172	1406	***12.2%***

Table 4. Effect of penalizing highways in attacker route selection.

When considering an extended model in which arbitrations of intercepts may be conducted, it is appropriate to consider the closing velocity of the vehicles at intercept and the resultant dwell time provided the searcher. A categorical look at the distribution of speed limits at the location of intercept is germane to this discussion. By aggregating all detections made over all searcher counts (50-800) within the same break-taking configuration (breaks enabled), we can develop a more complete picture of the benefit to dwell time imparted by forcing attackers onto city streets. See Figure 14.

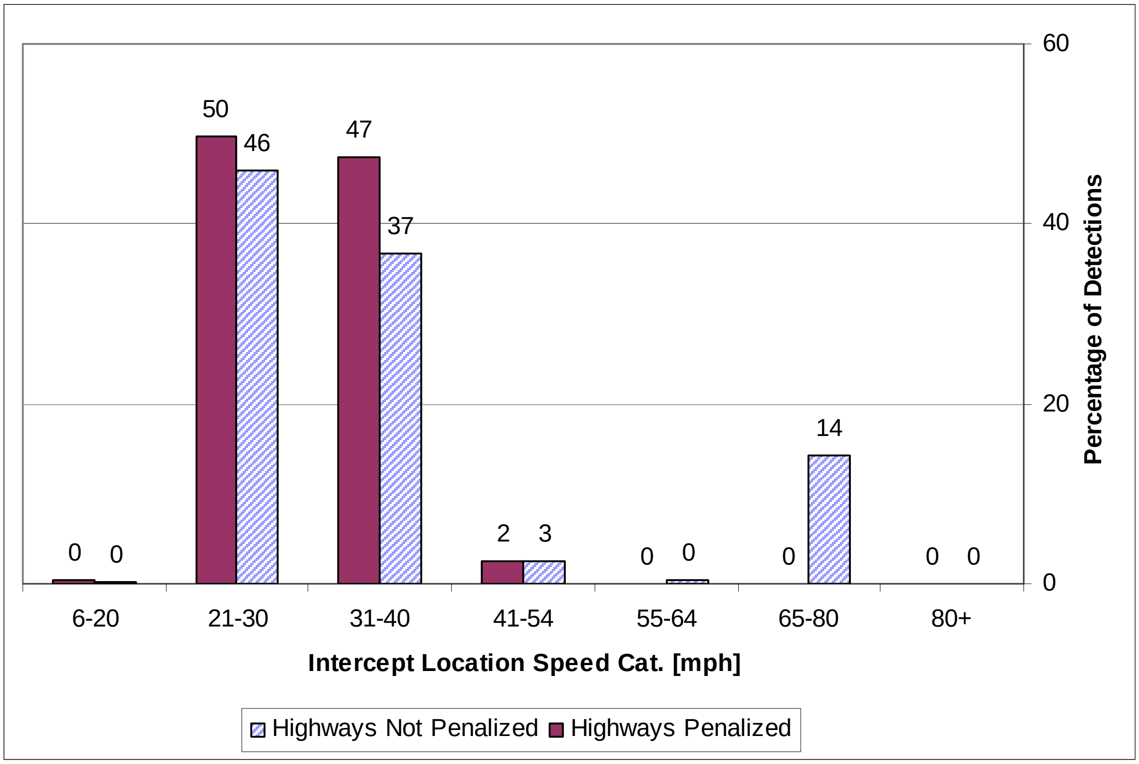


Figure 13. Distribution of road speed categories for intercepts without penalized highways

In comparing the aggregate number of detections depicted in Figure 13, we may wish to negate detections declared at relative closing speeds of greater than 80 m.p.h., which would represent a significant challenge to a mobile detector with current capabilities. This would force the exclusion of all intercepts declared on road segments with speeds in excess of 40 m.p.h. Reconsidering the aggregate data used to formulate the above comparison, a dramatic swing in effective detections declared occurs. The penalized highway model now shows a more distinct advantage due to the accumulation of lower speed intercepts. The shift in adjusted probability of detection over the aggregated dataset is depicted in Table 5.

Non-Penalized Highway Model

Raw Aggregate Detections	928 (11.6% Pd)
Excluded High Speed Detections	160
Detections Declared	768
Attacks Attempted	7970
Adjusted Aggregate Probability of Intercept	9.64%

Penalized Highway Model

Raw Aggregate Detections	1053 (15.1% Pd)
Excluded High Speed Detections	26
Detections Declared	1027
Attacks Attempted	6969
Adjusted Aggregate Probability of Intercept	14.7%

Table 5. Influence of excluding high-speed detections in both aggregated datasets

E. REACTION TIME CONSIDERATIONS

It is also desirable to generate possible reaction profiles from the data. Reaction profiles in this sense are numerical measures of how far from the target the attacker is when the intercept occurs, and also how much time is remaining on his attack route. Distance to the target at intercept is measured as a straight line distance, and is of concern when considering the extended effects radius presented by WMD detonations or releases. Ideally, a credible defensive structure would be developed that provides detection at a distance that precludes an early detonation from inflicting damage in the primary target area, and affords the maximum reaction time for law enforcement services to react and interdict the attacker. In the case of WMD threats, we should consider this safe distance in terms of *miles* from the target. A detection and intercept that finds the target to be within the lethal radius of the intercepted weapon may reduce the net effect of the weapon, but is certainly not a successful intercept.

Additionally, it is desirable to know how much time is remaining on the attacker's route when he is encountered. This time remaining translates into reaction time by law enforcement agencies. Ideally, a sensing agent would be able to reverse course in traffic and intercept the vehicle on-the-spot, but it is more likely the case that some additional effort in the form of a reaction team, secondary confirmation, target identification/confirmation maneuvering, or pursuit may be involved. It should be emphasized that any route timing generated by this simulation is an overly optimistic estimation of true road travel time. The simulation does not include traffic factors, nor does it factor in traffic control devices. Essentially, the route time generated reflects a speed-limit drive with green lights all the way.

Intuitively, straight line distance (SLD) and time to go (TTG) will both increase with number of searchers deployed, given that an intercept occurs. The data supports this, however the distribution of both within the attack radius is quite random, and only mildly influenced by increases in searcher numbers.

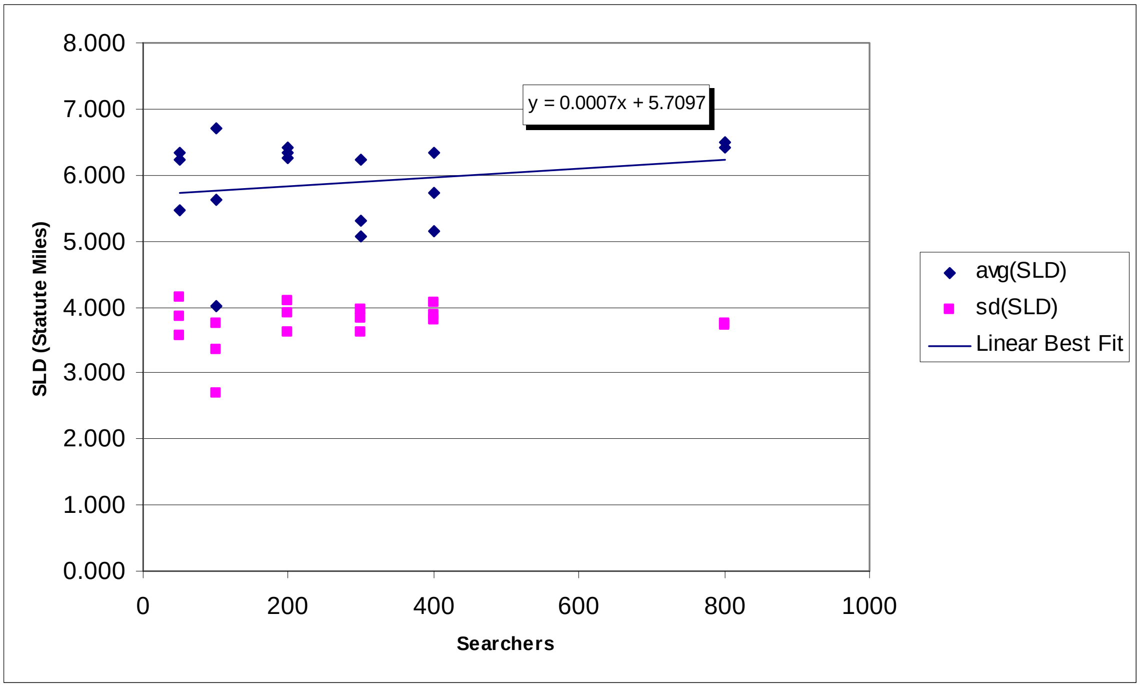


Figure 14. Straight Line Distance (SLD) to target at time of encounter for runs with highways penalized and no searcher breaks. Noise in low-searcher runs is driven by significantly lower overall intercepts in those configurations.

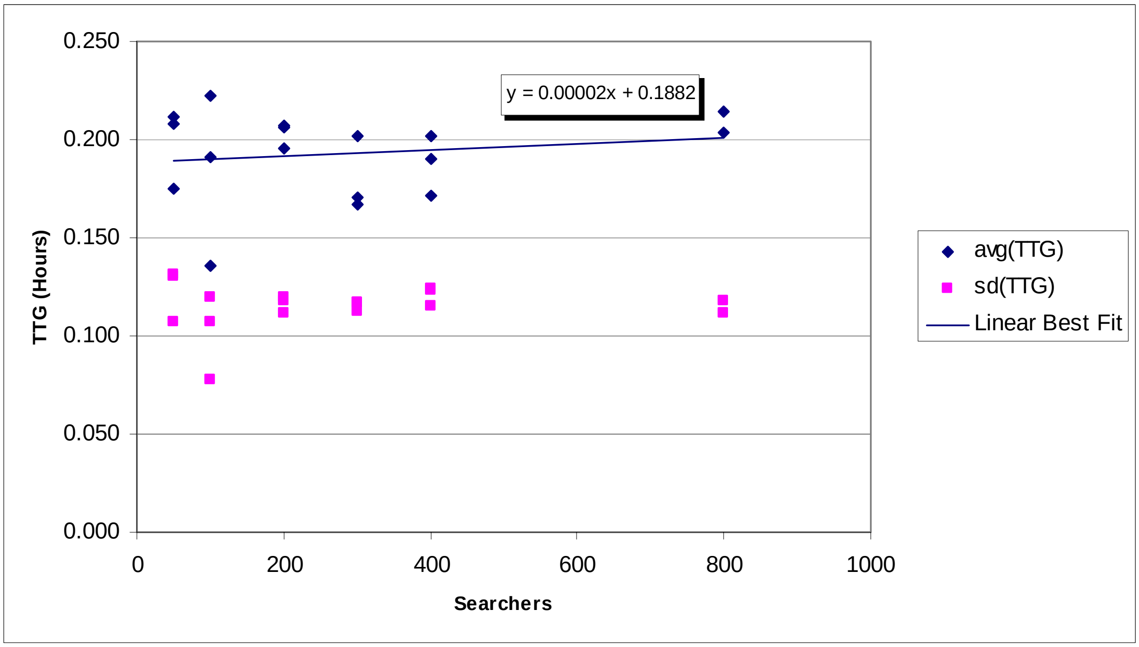


Figure 15. Time To Go (TTG) to target at time of encounter for same dataset.

In both of the reaction profile graphs above, the standard deviation of the data is on the order of half of the maximal, characteristic value for the response profiled, which is indicative of the extreme variability of the response. Specifically, the maximum SLD is approximately 10 miles, based on the geographically filtered start node set. The deviance in the SLD profile hovers around 4.0, indicating that predictions will vary wildly between zero and the maximal value of 10.0. A similar comparison can be drawn with the TTG profile, which carries a maximal value near 0.25 hours. The data represented in the SLD and TTG graphs is reproduced in Table 6. Each data point plotted in Figures 15 and 16 represents a run of fixed duration. Thus, configurations with fewer searchers will have lower accumulated intercepts and will show more dispersion. Grand averages are also presented as an aggregate measure.

		Hwy	**Take**	**Successful**	**Successful**				
City	**Searchers**	**Penalized**	**Breaks**	**Attacks**	**Intercepts**	**avg(SLD)**	**sd(SLD)**	**avg(TTG)**	**sd(TTG)**
						[miles]	[miles]	[hours]	[hours]
DC	50	TRUE	FALSE	375	14	5.462	3.561	0.175	0.107
DC	50	TRUE	FALSE	375	14	6.239	3.850	0.212	0.131
DC	50	TRUE	FALSE	379	13	6.345	4.141	0.208	0.131
DC	100	TRUE	FALSE	371	26	6.719	3.342	0.222	0.107
DC	100	TRUE	FALSE	365	27	4.022	3.756	0.136	0.120
DC	100	TRUE	FALSE	362	28	5.615	2.688	0.191	0.078
DC	200	TRUE	FALSE	350	55	6.251	4.101	0.196	0.120
DC	200	TRUE	FALSE	357	48	6.347	3.617	0.206	0.112
DC	200	TRUE	FALSE	349	61	6.425	3.897	0.207	0.118
DC	300	TRUE	FALSE	349	67	6.242	3.968	0.202	0.117
DC	300	TRUE	FALSE	333	78	5.069	3.613	0.167	0.112
DC	300	TRUE	FALSE	338	75	5.299	3.833	0.170	0.117
DC	400	TRUE	FALSE	318	104	5.161	3.789	0.171	0.115
DC	400	TRUE	FALSE	331	91	5.734	3.881	0.190	0.123
DC	400	TRUE	FALSE	329	97	6.325	4.064	0.202	0.124
DC	800	TRUE	FALSE	274	201	6.427	3.741	0.203	0.112
DC	800	TRUE	FALSE	275	199	6.504	3.730	0.214	0.118
					Grand Averages:	**5.893**		**0.193**	

Table 6. Reference data subset used for analysis of SLD and TTG

In general, the raw data suggest a nearly linear relationship between searchers employed and probability of detection generated in the band of searcher numbers explored. In the Washington D. C. area, penalizing highways within the beltway is a significant benefit to the searcher's efforts. Not only are more encounters produced, but a dramatic shift towards slower speed intercepts is recognized. Implications of these findings are more thoroughly developed in Chapter IV.

IV. CONCLUSIONS AND RECOMMENDATIONS

A. SIGNIFICANCE OF THE DETECTION MODEL

The overarching goal of this simulation development was to produce an effectiveness model for limited range mobile detectors applied in urban traffic schemes. This model shows that if no other means were used to augment the performance of randomly moving sensors that at least 150 sensors are required to generate a probability of detection greater than roughly 10% in the greater Washington D. C. area. This may seem like a paltry contribution from 150 advanced detectors, however, it must be realized that this baseline model only represents one mode of sensor employment, and presents fairly simple options for radically increasing performance. The low probability of detection generated by the simulation is principally the byproduct of three factors:

1. High ratio of potential attack routes to number of interdicting arcs patrolled.
2. Extremely limited detection range.
3. Sub-optimal searcher employment scheme in this base-case.

Before exploring possible performance enhancements to the search model, it should be noted that even fairly low probabilities of detection can serve as a significant deterrence factor or as a preventive measure inside a layered defensive structure. While a 10-20 percent chance of detection is ineffective in the face of repeated attacks or low-cost efforts on the part of the attacker, it does represent a significant deterrent to extraordinary and singular attempts by an attacker with limited capabilities. An attacking agent that is clearly exposed to huge risk and has devoted years of effort and finance to a singular WMD attack would likely find it difficult to reconcile with a 1-in-10 chance of failure in these efforts. Additionally, as mentioned in the description of layered security, the relatively small independent contribution from a detector scheme described herein adds significant power to a layered model through the power of accumulated probability of detection generated by the compounding effect of joint probabilities. Additionally, some means of detection and intervention must be preserved in any defensive system to counter a threat warning from within the territorial borders, for a suspected breach at a border entry point, or from a reliable, actionable item of domestic intelligence. Mobile detectors bring tremendous flexibility and adaptability to possible reaction schemes in these scenarios.

If we decide that screening effort within a major metropolitan area is necessary, mobile detectors become a front-running option. As mentioned previously, static defensive structures, especially when dispersed across large geographic areas can be observed, probed, and likely defeated by an adversary who is patient enough to wait for or produce a window opportunity. Inspecting the huge volume of vehicles entering or leaving a major metropolitan area has very low chance of serving as an adequate secondary measure. It is unsustainable, and easily recognized and countered or waited out. Mobile detector employment has been proposed and various performance enhancements will subsequently be presented.

B. MORE OPTIMIZED EMPLOYMENT SCHEMES

1. Smart Patrolling Patterns

The search patterns generated within the model are completely random, and although they serve as an effective base case, they do not represent savvy police work. Clearly, beats are established and higher profile areas defined by traffic density or crime statistics receive more attention than farm roads in suburbs outside the beltway. A reapportioning of searchers within beats would likely make a marked difference, along with emphasizing the patrol of arcs that are included in possible paths from outside the protected zone, as opposed to "drilling into" subsets of the graph that are not functionally part of any possible attack path. This would likely diminish the chances of detecting a rogue attack initiated from a location *within* the protected zone, but would provide dividends in increasing chances of detection for attacks initiated outside of the protected zone.

2. Cut Set Screening

If we consider a cut set of arcs on the graph to be smartly generated from street segments at an approximate fixed radial distance from the target area, we can manage screening on those arcs to produce large performance gains. In this scheme, cut sets are developed at radii from the target that balance the size of the cut set with the distance and time-to-go generated by a detection at that range. For example, a close inspection of the map structure may reveal that a reduced road density at a radius of approximately 7 miles from the target would produce a cut set size of 400 streets, and would generate an average reaction time of 20 minutes with 7 miles of distance from the target. [These numbers are hypothetical and are not implied to represent the D.C. dataset]. Stationary detector vehicles parked on arcs in this set would provide a ring of security with performance proportional to the ratio of arcs scanned. Covert deployment of the sensors in this case (via unmarked screening vehicles) would be an obvious step towards thwarting countermeasures. This implementation has additional benefits in reduced manpower requirements, as many vehicles could be parked unattended while nearby responders could act on linked threat information. This scheme has a better chance of stabilizing the SLD and TTG at detection times to a range actionable by response vehicles.

C. COUNTERMEASURES AND COUNTER-COUNTERMEASURES

Any robust attacker-defender model must take into consideration likely countermeasures by both sides. Certainly, an attacker who is aware of a detection and screening program will seek to improve the chances of success of a singular attack. WMD is not a game of long-run odds where the power of repetition can be invoked. Singular attacks will likely involve various contingency plans and at least modest countermeasures. The countermeasures chosen by the attacker will likely produce additional actions on the part of the defender. Contemporary examples of this behavior are found in electronic warfare in the form of radar jamming and counter-jamming radar waveforms, or in heat-seeking air-to-air missiles which frequently have guidance logic that specifically counters the flares meant to decoy them.

On initial inspection, various countermeasures on the part of the attacker are readily evident. A simple pilot-vehicle implementation has the potential to significantly reduce the already unlikely chance of an encounter with a searcher. In this scenario, a pilot or lead vehicle is employed to maneuver a few blocks ahead of the actual attack vehicle. Upon recognition of a sensor equipped vehicle (presumably a police cruiser), the pilot vehicle would direct the trailing attack vehicle to divert onto a side street to avoid the subsequent encounter with the patrol vehicle. Certainly, this could be countered with unmarked patrol vehicles, but at a likely cost of requiring additional vehicles and personnel on the roadways.

A second, devious, yet sadly plausible countermeasure involves a large-scale diversionary event. An event such as a large conventional explosive detonation in an opposing approach quadrant would likely divert many of the patrol assets away from the true attack path, allowing an attack vehicle a relatively uncontested entry into the primary target zone. Both of these relatively simplistic countermeasures need to be met with disciplined and methodical counter-countermeasures (CCM). Such CCM may involve increase use of dedicated, unmarked detection vehicles, many of which may be statically employed without additional personnel demands.

D. MODEL EXTENSIONS

The alternative, more optimized employment means delineated in this chapter already suggest viable extensions to the simulation model employed. The framework of the program would allow, with minor modification, the assignment of beats or patrols to the searchers to produce a more optimized patrol pattern. Additional structuring would be required to support an exploration of dynamically patrolling cut sets with patrol vehicles during the simulation run, but a well-defined low-count cut set derived by external map study could provide the framework of a first cut solution within the simulation. A selection of arcs (streets) from within this externally generated set could be flagged in the dataset in the simulation and used to generate statistical performance results.

Additional unique utilizations of mobile detectors also warrant further study. Suppose, for instance, a threat is levied against a particular target, the Capitol building, as an example. Mobile detectors could be used to screen and scan all parked vehicles within a critical defensive ring around this structure. The implementation would look something like a multiple-agent traveling salesman problem in which all streets (and parking structures) within a certain critical radius need to be swept by a slow-moving detector in order to clear the area. The time required to perform such a sweep would be a function of the number of sensors employed, routing decisions, and summary length of paths encircled.

Further exploration should also include response and intercept simulation. Currently, in this simplified model, encounter implies detection and detection implies intercept. Modeling of the detection physics for a nuclear, biological, or chemical attack is beyond the classification scope of an open simulation model, but reaction in traffic and any subsequent group behavior by the searcher "battalion" could be openly developed. For example, suppose a detection is declared in traffic. It is likely in city traffic that discriminating the vehicle which triggered the alert from the rest will require additional screening effort. The secondary screening might be produced by a rapid set of actions by the detecting unit (U-turn and pursuit for example), or by a coordinated response by other detection equipped vehicles. This secondary, confirmation reading should occur somewhere between the point of initial alert and the likely target area. The response by other agents and the resultant timeliness of the effort to interdict traffic flow and implement confirmation screening is directly linked to the probability of successful interdiction.

In summary, the cumulative deterrent and detection benefits of a layered defensive structure are clear. While low-deployment densities of mobile detecting agents do not generate large independent detection probabilities when employed in random search, the study does suggest that modest employment densities provide a very credible addition to the layered security model. Additionally, mobile detectors enable flexibility in response options and inject a largely stochastic patrol element into what is currently a static, observable, and vulnerable screening system. Additionally, as funds may become focused in this area, or as technology and production advances allow, deployment of sensors may become ubiquitous, in which case the power of the detection model becomes greatly magnified. It is not inconceivable that modest advances in detection hardware could facilitate small, linked, localizable, sensors mountable on all government and city vehicles, or even expand deployment to include citizenry who receive modest compensation for the government-owned black box they volunteer to have mounted in their vehicle's trunk. It must be restated that attacks that can produce unthinkable damage and loss must be met with highly creative and adaptive defensive measures. Mobile detectors certainly have a role in furthering this effort.

THIS PAGE INTENTIONALLY LEFT BLANK

V. APPENDIX A.: MAIN EXECUTABLE

The following code segment is provided as an example of the Java code structuring using simulation entities designed in SimKit. Additional classes do the support work of handling the graph and data structures.

```
package main;

import java.io.FileWriter;
import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import org.geotools.feature.Feature;
import org.geotools.feature.FeatureCollection;
import org.geotools.graph.structure.Graph;
import org.geotools.graph.structure.basic.BasicDirectedNode;

import simkit.Schedule;
import simkit.SimEntityBase;
import simkit.random.RandomVariate;
import simkit.random.RandomVariateFactory;
import util.InputHandler;
import util.algorithms.PathTool;
import util.algorithms.Reaching;
import util.algorithms.WeightedEdgeMap;
import util.entities.Searcher;
import util.intercept.InterceptReport;
import util.intercept.InterceptReportFactory;
import util.logging.BasicLogger;
import util.movement.AttackInstigator;
import util.movement.BasicGraphMover;
import util.movement.BasicInterceptMediator;
import util.movement.RandomSearchManager;
import util.movement.SearchLeader;
import util.viewer.CheapViewer;
import util.viewer.PathPlotter;

import com.vividsolutions.jts.geom.Coordinate;

/**
 * The main executable to run the simulation. It is set to use command line
 * arguments to set variables. Output can be either piped to System.out or
 * logged in a file based on command line settings.
 *
 * @version $Id: SimExecable.java,v 1.16 2007/02/12 19:26:25 jfhyink Exp $
 * @author J. F. Hyink
 *
 */
public class SimExecable extends SimEntityBase {

   @SuppressWarnings("unchecked")
   public static void main(String[] args) {

      // +++++++++++++++++++++++++++++++++++++++++++++++++++++
      //
      //                    Variables
      //
      // +++++++++++++++++++++++++++++++++++++++++++++++++++++

      // simulation entities
```

```java
GraphManager manager;
Graph graph;
WeightedEdgeMap edgeMap;
AttackInstigator instigator;
SearchLeader leader;
RandomSearchManager[] searchMoverManager;
BasicGraphMover[] searchMover;
Searcher[] searcher;
BasicInterceptMediator mediator;

// i/o variables
String cityName = null;
String targetFeatureID = null;
String searcherBaseFeatureID = null;
String datafileName = null;
URL sourceURL;
boolean pipeOut = true; // if pipeOut is true, output will be sent to
                  // screen, else it will be logged
String outputFilename = null;

// per-run variables
int speedPenaltyCutoffIndex = 999;
int numSearchers = 0;
boolean takesBreaks = true;
long seed = 0;

// simulation visualization variables
boolean startSetVisible = false;
boolean runtimeVisible = false;
boolean pathsAtCompletionVisible = false;

// text output modes
boolean consoleSugar = false;
boolean verbose = false;

// plotters and viewers
CheapViewer viewer = null;
PathPlotter plotter = null;

double runLength = 0.0; // hours of simulated time (set in input file)

// pull the input file and variables from command line args
if (args.length < 15) {
   System.err
         .println("expecting args in long form (see instructions)");
   System.err
         .println("dfile, name, tgt, base, rtime, srchrs, " +
               "spdCutoff, tbreaks, seed, cSugar, vrbse, "+
               "startVis, runVis, pathVis, pipeOut");
   System.exit(-1);
} else {
   datafileName = args[0];
   cityName = args[1];
   targetFeatureID = args[2];
   searcherBaseFeatureID = args[3];
   runLength = (new Double(args[4])).doubleValue();
   numSearchers = (new Integer(args[5])).intValue();
   speedPenaltyCutoffIndex = (new Integer(args[6])).intValue();
   takesBreaks = (args[7].charAt(0) == 't') ? true : false;
   seed = Long.parseLong(args[8]);
   consoleSugar = (args[9].charAt(0) == 't') ? true : false;
   verbose = (args[10].charAt(0) == 't') ? true : false;
   startSetVisible = (args[11].charAt(0) == 't') ? true : false;
   runtimeVisible = (args[12].charAt(0) == 't') ? true : false;
   pathsAtCompletionVisible = (args[13].charAt(0) == 't') ? true
         : false;
   pipeOut = (args[14].charAt(0) == 't') ? true : false;
   if (args.length > 15)
      outputFilename = args[15];
}
```

```
// ******************************
// Set the seed of the RandomVariateFactory
// ******************************

RandomVariateFactory.getDefaultRandomNumber().setSeed(seed);

// load a shapefile into data storage
InputHandler handler = new InputHandler();
String[] source;
source = new String[] { datafileName };
sourceURL = handler.getURLtoFile(source);

// this "makes" the graph
manager = new GraphManager(sourceURL, speedPenaltyCutoffIndex);
manager.setVerbose(verbose);
edgeMap = manager.getEdgeMap();
graph = manager.getFGG().getGraph();

// build check:
int edges = graph.getEdges().size();
if (consoleSugar)
   System.err.println("\nedges in graph constructed: " + edges);

// show the map at build-time if visible is set
if (startSetVisible) {
   viewer = new CheapViewer(manager.getCRS());
   viewer.showCollection(manager.getFeatureCollection(),
         CheapViewer.GRAY);
   // viewer.pauseForReturn();
}

// set up a PathTool utility used for routing and reaching stuff
PathTool pathTool = new PathTool(edgeMap, graph);

// anchor the homeBase and target Features
BasicDirectedNode homeNode = null, targetNode = null;
targetNode = (BasicDirectedNode) manager.getEdge(targetFeatureID)
      .getOutNode();
homeNode = (BasicDirectedNode) manager.getEdge(searcherBaseFeatureID)
      .getOutNode();

// show reaching stats on graph built (principally for QA/error
// checking)
int unreachableSetSize = Reaching.unreachableNodes(graph, homeNode)
      .size();
int isolatedSetSize = Reaching.isolatedNodes(graph, homeNode).size();
if (consoleSugar) {
   System.err.println("Unreachable set size: " + unreachableSetSize);
   System.err.println("Isolated set size: " + isolatedSetSize);
}

// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
//
// find legitimate starting nodes for attack based on intersections
// with a circle that fits within the boundaries of the map
//
// +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FeatureCollection starters = null;
ArrayList<Feature> startingFeatures = new ArrayList<Feature>();
HashSet<BasicDirectedNode> startingNodes = null;

// find starting nodes graphically
Coordinate center = null;
center = manager.getFeatureCollection().getBounds().centre();
double maxWidth = manager.getFeatureCollection().getBounds().getWidth();
double maxHeight = manager.getFeatureCollection().getBounds()
      .getHeight();
double minDemension = (maxWidth < maxHeight) ? maxWidth : maxHeight;
// 95% of minimum dimension's half-width for radius
double attackRadius = minDemension * (0.5) * (0.95);
```

```java
try {
   starters = manager.getFeaturesAtRadius(center, attackRadius);
   startingFeatures.addAll(starters);
   startingNodes = manager.getInNodesFromFeatures(startingFeatures);
   if (consoleSugar) {
      System.err.println("Attack node start points generated: "
            + startingNodes.size());
   }
   if (startSetVisible) {
      viewer.addLayerOnTop(starters, CheapViewer.RED);
      if (consoleSugar)
         System.err.println("hit return to continue run.");
      viewer.pauseForReturn();

   }
} catch (IOException e) {
   e.printStackTrace();
}

// ********************************************
// set up remaining standing entities
// ********************************************

InterceptReportFactory interceptReportFactory =
   new InterceptReportFactory(manager, pathTool);
mediator = new BasicInterceptMediator(manager, interceptReportFactory);
mediator.setVisual(runtimeVisible);
mediator.setVerbose(verbose);
leader = new SearchLeader(graph);
instigator = new AttackInstigator(startingNodes, targetNode, graph,
      edgeMap, mediator);
instigator.setConsoleSugar(consoleSugar);

// enable the attackInstigator to create penalized routes if flag
// (999) is NOT set
if (speedPenaltyCutoffIndex != 999) {
   instigator.setPenaltyMap(manager.getPenaltyMap());
} else { // turn it off
   instigator.setUsePenaltyMap(false);
}

// *******************************************
// set up listeners for standing entities
// *******************************************

// this will allow the instigator to "scramble" the searchers
instigator.addSimEventListener(leader);

// set up RandomVariates
RandomVariate breakDuration = RandomVariateFactory.getInstance(
      "Uniform", new Object[] { new Double(0.2), new Double(1.0) });
RandomVariate timeBetweenBreaks = RandomVariateFactory.getInstance(
      "Uniform", new Object[] { new Double(0.5), new Double(1.0) });
double proportionOnBreakAtStart = 0.55;

// *********************
// set up a logger
// *********************

FileWriter w = null;
if (!pipeOut) { // if w is left null, logger will instantiate to got to
            // System.out
   try {
      w = new FileWriter(outputFilename);
   } catch (IOException e) {
      e.printStackTrace();
   }
}
BasicLogger logger = new BasicLogger(w, manager);
// logger hears reports from report factory and the instigator
```

```
interceptReportFactory.addPropertyChangeListener(logger);
instigator.addPropertyChangeListener(logger);

// *********************
// set up a path plotter if visible is set
// *********************
if (pathsAtCompletionVisible) {
   plotter = new PathPlotter(manager);
   mediator.addPropertyChangeListener(plotter);
   instigator.addPropertyChangeListener(plotter);
}

// **************************
// set up the run
// **************************

// initialize the appropriate number of searchers
int searchers = numSearchers;
searcher = new Searcher[searchers];
searchMover = new BasicGraphMover[searchers];
searchMoverManager = new RandomSearchManager[searchers];

// initialize movers and searchers
for (int i = 0; i < searchers; i++) {
   searcher[i] = new Searcher("serno " + i);
   searchMover[i] = new BasicGraphMover(searcher[i], edgeMap);
   searchMoverManager[i] = new RandomSearchManager(manager.getFGG()
         .getGraph(), homeNode, pathTool);

   searchMoverManager[i].setupBreaks(timeBetweenBreaks, breakDuration,
         proportionOnBreakAtStart);
   searchMoverManager[i].setTakesBreaks(takesBreaks);
   searchMoverManager[i].setUTurnMaker(false);

   // set up listeners for search entities
   searchMover[i].addSimEventListener(searchMoverManager[i]);
   searchMoverManager[i].addSimEventListener(searchMover[i]);

   // connect to searchLeader
   leader.addSimEventListener(searchMoverManager[i]);
   searchMoverManager[i].addSimEventListener(leader);

   // connect mover to mediator
   searchMover[i].addSimEventListener(mediator);
   mediator.addSimEventListener(searchMover[i]);
}

// send the run variables to the logger
logger.setCityName(cityName);
logger.setSearchersUsed(searchers);
logger.setTakesBreaks(takesBreaks);
if (speedPenaltyCutoffIndex != 999)
   logger.setPenalizedHighways(true);
else
   logger.setPenalizedHighways(false);

Schedule.reset();
Schedule.stopAtTime(runLength);
Schedule.setVerbose(verbose);
Schedule.setReallyVerbose(false);
Schedule.startSimulation();

// log results
logger.endRun();
if (consoleSugar) {
   System.err.println(
         "\n\t**********" +
         "\n\t*        *" +
         "\n\t*  DONE  *" +
         "\n\t*        *" +
         "\n\t**********");
```

```
      }

      // ****************
      // summary jobs
      // ****************

      if (runtimeVisible) {
         // kill the realtime viewer
         viewer.getFrame().dispose();
      }

      if (consoleSugar) {// show the number of isolated entries for QA
         int isoEntries = 0;
         for (int i = 0; i < searchMoverManager.length; i++) {
            isoEntries += searchMoverManager[i].getNumEntriesToIsolated();
         }
         System.err.println("Number of entries to isolated areas " +
               "requiring reset: "  + isoEntries);
      }

      if (verbose) {// show the interceps on the console
         HashSet<InterceptReport> reportSet = new HashSet<InterceptReport>();
         reportSet.addAll(mediator.getReports());
         Iterator<InterceptReport> i = reportSet.iterator();
         System.out
            .println("\n ****************** intercepts******************"
                     + "\n total intercepts: " + reportSet.size());
         int count = 0;
         double cumulativeDist = 0.0;
         while (i.hasNext()) {
            InterceptReport current = i.next();
            System.out.println("\n**** intercept report: " + ++count);
            System.out.println(current);
            cumulativeDist += current.getSlantRangeToTarget();
         }
         System.out.println("average intercept range: " + cumulativeDist
               / count);
      }

      if (pathsAtCompletionVisible) {
         if (PathPlotter.isRunning()) {
            PathPlotter.getViewer().showCollection(
                  manager.getFeatureCollection(), CheapViewer.GRAY);
            plotter.showSuccessfulAttacks();
            plotter.showInterceptedAttacks();
            PathPlotter.refresh();
         }
      }
      // ***********************
      // cleanup
      // ***********************

      // kill the open writer thread in the logger
      try {
         if (!pipeOut)
            logger.getWriter().close();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }

   protected static double convertStatuteMilesToDegrees(double statuteMiles) {
      double nauticalMiles = statuteMiles * 1.15077945;
      double degrees = nauticalMiles / 60.0;
      return degrees;
   }

}
```

VI. APPENDIX B.: RAW DATA

The data used for analysis is presented below. Each row represents a run of 200 simulated hours at the given set point. Various runs were executed with 3000 searchers employed. This number of searchers is well outside the normal bandwidth of employment options (0 – 800), and the results were excluded from the logistic model and analysis.

Configuration				Results								Speed Category Distribution of Intercepts							
City	Searchers	Hwy Pen	TakeBreak	Successful Attacks	Successful Intercepts	p(S.Attk)	p(S.Intcpt)	avSLD	sdSLD	avTTG	sdTTG	cat 1	cat 2	cat 3	cat 4	cat 5	cat 6	cat 7	cat 8
DC	50	FALSE	TRUE	437	15	0.967	0.033	4.681	4.355	0.137	0.122	0	2	0	2	4	7	0	0
DC	50	FALSE	TRUE	441	9	0.980	0.020	4.580	2.751	0.152	0.079	0	1	0	0	2	6	0	0
DC	50	FALSE	TRUE	440	12	0.973	0.027	5.616	4.692	0.173	0.140	0	1	0	0	3	8	0	0
DC	100	FALSE	TRUE	434	12	0.973	0.027	5.426	3.961	0.192	0.136	0	0	0	0	6	6	0	0
DC	100	FALSE	TRUE	437	20	0.956	0.044	5.183	4.571	0.171	0.141	0	1	0	0	6	13	0	0
DC	100	FALSE	TRUE	434	26	0.943	0.057	5.812	3.835	0.166	0.108	0	6	0	0	14	6	0	0
DC	200	FALSE	TRUE	422	49	0.896	0.104	5.254	3.650	0.163	0.107	0	6	0	2	18	23	0	0
DC	200	FALSE	TRUE	418	49	0.895	0.105	5.025	3.349	0.154	0.095	0	9	0	0	17	23	0	0
DC	200	FALSE	TRUE	422	47	0.900	0.100	5.388	4.287	0.167	0.130	0	5	0	1	13	28	0	0
DC	300	FALSE	TRUE	415	50	0.892	0.108	5.583	4.386	0.166	0.137	0	8	0	3	15	24	0	0
DC	300	FALSE	TRUE	407	67	0.859	0.141	5.955	3.739	0.191	0.117	0	7	1	2	26	31	0	0
DC	300	FALSE	TRUE	412	55	0.882	0.118	5.214	4.116	0.159	0.124	0	7	0	1	19	28	0	0
DC	400	FALSE	TRUE	392	90	0.813	0.187	5.890	3.958	0.178	0.121	0	13	1	3	37	36	0	0
DC	400	FALSE	TRUE	394	84	0.824	0.176	5.227	3.957	0.165	0.122	0	9	0	3	33	39	0	0
DC	400	FALSE	TRUE	407	66	0.860	0.140	5.283	3.445	0.165	0.103	0	10	0	0	22	34	0	0
DC	800	FALSE	TRUE	363	146	0.713	0.287	5.602	3.932	0.163	0.113	0	29	1	3	58	55	0	0
DC	800	FALSE	TRUE	367	131	0.737	0.263	5.476	4.021	0.170	0.124	0	19	0	4	47	60	1	0
DC	50	TRUE	TRUE	372	15	0.961	0.039	3.265	3.458	0.118	0.116	0	0	0	1	2	12	0	0
DC	50	TRUE	TRUE	374	13	0.966	0.034	5.502	4.283	0.189	0.146	0	0	0	0	4	8	1	0
DC	50	TRUE	TRUE	378	8	0.979	0.021	6.145	4.877	0.180	0.126	0	0	0	0	7	1	0	0
DC	100	TRUE	TRUE	375	22	0.945	0.055	6.629	3.973	0.216	0.129	0	0	0	0	10	12	0	0
DC	100	TRUE	TRUE	371	22	0.944	0.056	6.500	3.988	0.201	0.114	0	0	0	0	12	10	0	0
DC	100	TRUE	TRUE	367	25	0.936	0.064	6.547	3.665	0.203	0.101	0	0	0	0	11	14	0	0
DC	200	TRUE	TRUE	350	56	0.862	0.138	6.064	3.812	0.189	0.109	0	0	0	1	33	21	1	0
DC	200	TRUE	TRUE	362	47	0.885	0.115	6.465	3.623	0.200	0.104	0	0	0	2	22	23	0	0
DC	200	TRUE	TRUE	360	42	0.896	0.104	5.829	3.227	0.179	0.087	0	0	0	2	16	24	0	0
DC	300	TRUE	TRUE	349	59	0.855	0.145	5.994	3.875	0.184	0.117	0	0	0	1	32	26	0	0
DC	300	TRUE	TRUE	343	68	0.835	0.165	6.262	3.882	0.207	0.123	0	0	0	0	40	28	0	0
DC	300	TRUE	TRUE	345	67	0.837	0.163	5.860	3.920	0.180	0.110	0	0	0	2	26	39	0	0
DC	400	TRUE	TRUE	327	92	0.780	0.220	6.101	3.892	0.205	0.123	0	0	0	0	43	49	0	0
DC	400	TRUE	TRUE	335	81	0.805	0.195	5.266	3.896	0.165	0.110	0	0	0	2	37	42	0	0
DC	400	TRUE	TRUE	328	101	0.765	0.235	6.359	3.969	0.206	0.124	0	0	0	3	42	55	1	0
DC	800	TRUE	TRUE	286	170	0.627	0.373	6.271	3.879	0.199	0.115	0	0	0	5	87	78	0	0
DC	800	TRUE	TRUE	294	165	0.641	0.359	6.232	3.813	0.198	0.113	0	0	0	7	75	82	1	0
DC	50	FALSE	FALSE	432	21	0.954	0.046	6.867	3.922	0.212	0.122	0	3	0	0	10	8	0	0
DC	50	FALSE	FALSE	440	8	0.982	0.018	7.258	3.224	0.229	0.118	0	2	0	0	4	2	0	0
DC	50	FALSE	FALSE	438	8	0.982	0.018	4.491	3.944	0.134	0.131	0	2	0	0	2	4	0	0
DC	100	FALSE	FALSE	435	23	0.950	0.050	4.905	3.758	0.159	0.116	0	2	0	0	9	12	0	0
DC	100	FALSE	FALSE	434	26	0.943	0.057	6.515	4.123	0.200	0.134	0	5	0	1	8	12	0	0
DC	100	FALSE	FALSE	429	23	0.949	0.051	4.719	3.586	0.135	0.094	0	6	0	0	7	10	0	0
DC	200	FALSE	FALSE	413	56	0.881	0.119	5.020	4.163	0.167	0.133	0	3	0	1	20	32	0	0
DC	200	FALSE	FALSE	423	43	0.908	0.092	4.896	3.499	0.160	0.112	0	3	0	0	17	23	0	0
DC	300	FALSE	FALSE	407	67	0.859	0.141	6.280	3.979	0.197	0.121	0	5	0	3	24	35	0	0
DC	300	FALSE	FALSE	412	65	0.864	0.136	6.100	4.235	0.186	0.140	0	17	0	0	24	23	1	0
DC	300	FALSE	FALSE	408	62	0.868	0.132	6.400	4.021	0.200	0.122	0	6	0	1	31	24	0	0
DC	400	FALSE	FALSE	390	94	0.806	0.194	5.704	3.831	0.180	0.117	0	10	0	1	32	51	0	0
DC	400	FALSE	FALSE	395	90	0.814	0.186	6.044	4.089	0.186	0.126	0	11	0	1	39	38	1	0
DC	400	FALSE	FALSE	397	91	0.814	0.186	5.218	4.053	0.163	0.127	0	16	0	2	28	45	0	0
DC	800	FALSE	FALSE	346	165	0.677	0.323	5.408	4.039	0.168	0.122	0	21	0	1	59	83	1	0
DC	800	FALSE	FALSE	340	173	0.663	0.337	5.914	3.829	0.181	0.115	0	30	0	7	61	74	1	0
DC	3000	FALSE	FALSE	188	482	0.281	0.719	6.908	4.153	0.208	0.127	0	73	0	12	210	181	6	0
DC	3000	FALSE	FALSE	206	445	0.316	0.684	6.952	4.181	0.213	0.130	0	63	0	9	198	170	5	0
DC	50	TRUE	FALSE	375	14	0.964	0.036	5.462	3.561	0.175	0.107	0	0	0	0	9	5	0	0
DC	50	TRUE	FALSE	375	14	0.964	0.036	6.239	3.850	0.212	0.131	0	0	0	0	7	7	0	0
DC	50	TRUE	FALSE	379	13	0.967	0.033	6.345	4.141	0.208	0.131	0	0	0	0	9	4	0	0
DC	100	TRUE	FALSE	371	26	0.935	0.065	6.719	3.342	0.222	0.107	0	0	0	0	13	13	0	0
DC	100	TRUE	FALSE	365	27	0.931	0.069	4.022	3.756	0.136	0.120	0	0	0	1	9	17	0	0
DC	100	TRUE	FALSE	362	28	0.928	0.072	5.615	2.688	0.191	0.078	0	0	0	0	11	17	0	0
DC	200	TRUE	FALSE	350	55	0.864	0.136	6.251	4.101	0.196	0.120	0	0	0	3	25	27	0	0
DC	200	TRUE	FALSE	357	48	0.881	0.119	6.347	3.617	0.206	0.112	0	0	0	1	27	20	0	0
DC	200	TRUE	FALSE	349	61	0.851	0.149	6.425	3.897	0.207	0.118	0	0	0	2	25	34	0	0
DC	300	TRUE	FALSE	349	67	0.839	0.161	6.242	3.968	0.202	0.117	0	0	0	3	23	41	0	0
DC	300	TRUE	FALSE	333	78	0.810	0.190	5.069	3.613	0.167	0.112	0	0	0	1	35	42	0	0
DC	300	TRUE	FALSE	338	75	0.818	0.182	5.299	3.833	0.170	0.117	0	0	0	5	34	36	0	0
DC	400	TRUE	FALSE	318	104	0.754	0.246	5.161	3.789	0.171	0.115	0	0	0	2	42	60	0	0
DC	400	TRUE	FALSE	331	91	0.784	0.216	5.734	3.881	0.190	0.123	0	0	0	3	42	46	0	0
DC	400	TRUE	FALSE	329	97	0.772	0.228	6.325	4.064	0.202	0.124	0	0	0	1	35	61	0	0
DC	800	TRUE	FALSE	274	201	0.577	0.423	6.427	3.741	0.203	0.112	0	0	0	1	92	108	0	0
DC	800	TRUE	FALSE	275	199	0.580	0.420	6.504	3.730	0.214	0.118	0	0	0	3	88	107	1	0
DC	3000	TRUE	FALSE	117	571	0.170	0.830	7.572	3.822	0.239	0.119	0	0	0	13	305	250	3	0
DC	3000	TRUE	FALSE	131	533	0.197	0.803	7.417	3.737	0.236	0.117	0	0	0	11	297	224	1	0

Table 7. Raw Data

LIST OF REFERENCES

Buss, Arnold H., and Paul Sanchez. "Building Complex Models with LEGOs (Listener Event Graph Objects)." *Proceedings of the 2002 Winter Simulation Conference* (2002).

Edmunds, Thomas A. 1994. A Markov Chain Model for Evaluating the Effectiveness of Randomized Surveillance Procedures. Lawrence Livermore National Laboratories

Edmunds, Thomas A., Gansemer, J., Nelson, K., Beauchamp, B., Lange, D., Dooher, B., Coty, J., Wheeler, R. 2006. Mobile Detection Systems for Urban Defense. Lawrence Livermore National Laboratories. (For Official Use Only)

Maersk Line Company Website. News, 1 SEP 2006. http://www.maerskline.com/link/?page=news&path=/news/news20060901, Last accessed: 24 JAN 2007

National Atlas. Raw Data Downloads, accessed 1 FEB 2007. http://nationalatlas.gov/atlasftp.html, Last accessed: 24 JAN 2007

NAVTEQ NAVSTREETS dataset Version 3.3.0, ESRI Corporation, through Lawrence Livermore National Labs

GeoTools, the open source Java GIS toolkit. http://docs.codehaus.org/display/GEOTOOLS/Home

SimKit, Discrete Event Simulation package, http://diana.nps.edu/Simkit/, Last accessed: 25 FEB 2007

Wagner, D., W. C. Mylander, and T. Sanders. 1999. *Naval Operations Analysis.* Naval Institute Press

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Don Brutzman
   MOVES Institute, Naval Postgraduate School
   Monterey, CA

4. Dr. Bob Atwell
   Institute For Defense Analyses
   Strategy, Forces, & Resources Division
   Alexandria, VA

5. Don Gaver
   Operations Research Department, Naval Postgraduate School
   Monterey, CA

6. Thomas A. Edmunds
   Lawrence Livermore National Laboratory
   Electronics Engineering Technology Division
   Livermore, CA